CHAPTER 9

Traditional Spanning Tree Protocol

Previous chapters covered robust network designs where redundant links are used between switches. Although this increases the network availability, it also opens up the possibility for conditions that would impair the network. In a Layer 2 switched network, preventing bridging loops from forming over redundant paths is important. Spanning Tree Protocol (STP) was designed to monitor and control the Layer 2 network so that a loop-free topology is maintained.

This chapter discusses the theory and operation of the STP. More specifically, the original, or traditional, STP is covered, as defined in IEEE 802.1D. Several chapters explain STP topics in this book. Here is a brief roadmap so you can chart a course:

- **Chapter 9, "Traditional Spanning Tree Protocol"**—Covers the theory of IEEE 802.1D
- **Chapter 10, "Spanning Tree Configuration"**—Covers the configuration commands needed for IEEE 802.1D
- **Chapter 11, "Protecting the Spanning Tree Protocol Topology"**—Covers the features and commands to filter and protect a converged STP topology from conditions that could destabilize it
- **Chapter 12, "Advanced Spanning Tree Protocol"**—Covers the newer 802.1w and 802.1s enhancements to STP, allowing more scalability and faster convergence

"Do I Know This Already?" Quiz

The purpose of the "Do I Know This Already?" quiz is to help you decide whether you need to read the entire chapter. If you already intend to read the entire chapter, you do not necessarily need to answer these questions now.

The 12-question quiz, derived from the major sections in the "Foundation Topics" portion of the chapter, helps you determine how to spend your limited study time.

Table 9-1 outlines the major topics discussed in this chapter and the "Do I Know This Already?" quiz questions that correspond to those topics.

Table 9-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundation Topics Section	Questions Covered in This Section	Score
IEEE 802.1D	1–10	
Types of STP	11–12	
Total Score		

CAUTION The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark this question wrong. Giving yourself credit for an answer you correctly guess skews your self- assessment results and might give you a false sense of security.

1. How is a bridging loop best described?

   a. A loop formed between switches for redundancy

   b. A loop formed by the Spanning Tree Protocol

   c. A loop formed between switches where frames circulate endlessly

   d. The round-trip path a frame takes from source to destination

2. Which of these is one of the parameters used to elect a Root Bridge?

   a. Root Path Cost

   b. Path Cost

   c. Bridge Priority

   d. BPDU revision number

3. If all switches in a network are left at their default STP values, which one of the following is not true?

   a. The Root Bridge will be the switch with the lowest MAC address.

   b. The Root Bridge will be the switch with the highest MAC address.

   c. One or more switches will have a Bridge Priority of 32,768.

   d. A secondary Root Bridge will be present on the network.

4. Configuration BPDUs are originated by which of the following?

    a. All switches in the STP domain

    b. Only the Root Bridge switch

    c. Only the switch that detects a topology change

    d. Only the secondary Root Bridge when it takes over

5. Which of these is the single most important design decision to be made in a network running STP?

    a. Removing any redundant links

    b. Making sure all switches run the same version of IEEE 802.1D

    c. Root Bridge placement

    d. Making sure all switches have redundant links

6. What happens to a port that is neither a Root Port nor a Designated Port?

    a. It is available for normal use.

    b. It can be used for load balancing.

    c. It is put into the Blocking state.

    d. It is disabled.

7. What is the maximum number of Root Ports that a Catalyst switch can have?

    a. 1

    b. 2

    c. Unlimited

    d. None

8. What mechanism is used to set STP timer values for all switches in a network?

    a. Configuring the timers on every switch in the network.

    b. Configuring the timers on the Root Bridge switch.

    c. Configuring the timers on both primary and secondary Root Bridge switches.

    d. The timers can't be adjusted.

9. MAC addresses can be placed into the CAM table, but no data can be sent or received if a switch port is in which of the following STP states?

    a. Blocking

    b. Forwarding

    c. Listening

    d. Learning

10. What is the default “hello” time for IEEE 802.1D?

- **a.** 1 second
- **b.** 2 seconds
- **c.** 30 seconds
- **d.** 60 seconds

11. Which of the following is the Spanning Tree Protocol defined in the IEEE 802.1Q standard?

- **a.** PVST
- **b.** CST
- **c.** EST
- **d.** MST

12. If a switch has 10 VLANs defined and active, how many instances of STP will run using PVST+ versus CST?

- **a.** 1 for PVST+, 1 for CST
- **b.** 1 for PVST+, 10 for CST
- **c.** 10 for PVST+, 1 for CST
- **d.** 10 for PVST+, 10 for CST

You can find the answers to the quiz in Appendix A, “Answers to Chapter ’Do I Know This Already?’ Quizzes and Q&A Sections.” The suggested choices for your next step are as follows:

- **7 or less overall score**—Read the entire chapter. This includes the “Foundation Topics,” “Foundation Summary,” and “Q&A” sections.
- **8–10 overall score**—Begin with the “Foundation Summary” section and then follow up with the “Q&A” section at the end of the chapter.
- **11 or more overall score**—If you want more review on these topics, skip to the “Foundation Summary” section and then go to the “Q&A” section at the end of the chapter. Otherwise, move to Chapter 10.

Foundation Topics

IEEE 802.1D Overview

A robust network design not only includes efficient transfer of packets or frames, but also considers how to recover quickly from faults in the network. In a Layer 3 environment, the routing protocols in use keep track of redundant paths to a destination network so that a secondary path can be used quickly if the primary path fails. Layer 3 routing allows many paths to a destination to remain up and active, and allows load sharing across multiple paths.

In a Layer 2 environment (switching or bridging), however, no routing protocols are used, and active redundant paths are neither allowed nor desirable. Instead, some form of bridging provides data transport between networks or switch ports. The Spanning Tree Protocol (STP) provides network link redundancy so that a Layer 2 switched network can recover from failures without intervention in a timely manner. The STP is defined in the IEEE 802.1D standard.

STP is discussed in relation to the problems it solves in the sections that follow.

Bridging Loops

Recall that a Layer 2 switch mimics the function of a transparent bridge. A transparent bridge must offer segmentation between two networks while remaining transparent to all the end devices connected to it. For the purpose of this discussion, consider a two-port Ethernet switch and its similarities to a two-port transparent bridge.

A transparent bridge (and the Ethernet switch) must operate as follows:

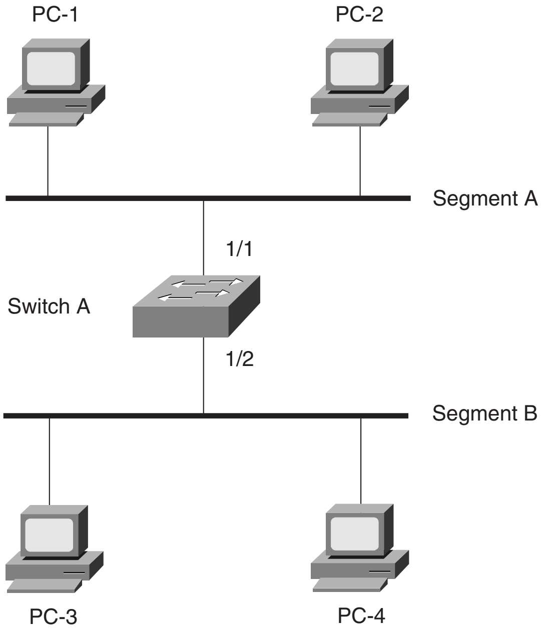
- The bridge has no initial knowledge of any end device's location; therefore, the bridge must "listen" to frames coming into each of its ports to figure out on which network each device resides. The bridge assumes that a device using the source MAC address is located behind the port that the frame arrives on. As the listening process continues, the bridge builds a table that correlates source MAC addresses with the Bridge Port numbers where they were detected.

  The bridge constantly can update its bridging table upon detecting the presence of a new MAC address or upon detecting a MAC address that has changed location from one Bridge Port to another. The bridge then can forward frames by looking at the destination MAC address, looking up that address in the bridge table, and sending the frame out the port where the destination device is known to be located.

- If a frame arrives with the broadcast address as the destination address, the bridge must forward, or flood, the frame out all available ports. However, the frame is not forwarded out the port that initially received the frame. In this way, broadcasts can reach all available Layer 2 networks. A bridge segments only collision domains—it does not segment broadcast domains.
- If a frame arrives with a destination address that is not found in the bridge table, the bridge cannot determine which port to forward the frame to for transmission. This type of frame is known as an *unknown unicast*. In this case, the bridge treats the frame as if it were a broadcast and floods it out all remaining ports. When a reply to that frame is overheard, the bridge can learn the location of the unknown station and can add it to the bridge table for future use.
- Frames forwarded across the bridge cannot be modified by the bridge itself. Therefore, the bridging process is effectively *transparent*.

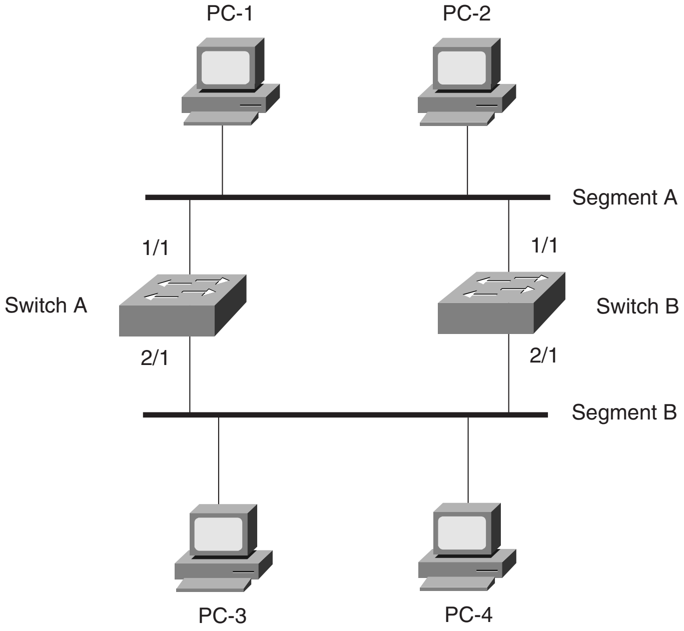
Bridging or switching in this fashion works well. Any frame forwarded, whether to a known or unknown destination, is forwarded out the appropriate port or ports so that it is likely to be received successfully at the end device. Figure 9-1 shows a simple two-port switch functioning as a bridge, forwarding frames between two end devices. However, this network design offers no additional links or paths for redundancy if the switch or one of its links fails. In that case, the networks on either side of the bridge would become isolated from each other.

Figure 9-1 *Transparent Bridging with a Switch*



To add some redundancy, you can add a second switch between the two original network segments, as shown in Figure 9-2. Now, two switches offer the transparent bridging function in parallel. In theory, a single switch or a single link can fail without causing end-to-end connectivity to fail.

Figure 9-2 *Redundant Bridging with Two Switches*



Consider what happens when PC-1 sends a frame to PC-4. For now, assume that both PC-1 and PC-4 are known to the switches and are in their address tables. PC-1 sends the frame onto network Segment A. Switch A and switch B both receive the frame on their 1/1 ports. Because PC-4 already is known to the switches, the frame is forwarded out ports 2/1 on each switch onto Segment B. The end result is that PC-4 receives two copies of the frame from PC-1. This is not ideal, but it is not disastrous, either.

Now, consider the same process of sending a frame from PC-1 to PC-4. This time, however, neither switch knows anything about the location of PC-1 or PC-4. PC-1 sends the frame to PC-4 by placing it on Segment A. The sequence of events is as follows:

Step 1 Both Switch A and switch B receive the frame on their 1/1 ports. Because the MAC address of PC-1 has not yet been seen or recorded, each switch records PC-1's MAC address in its address table along with the receiving port number, 1/1. From this information, both switches infer that PC-1 must reside on Segment A.

Step 2 Because the location of PC-4 is unknown, both switches correctly decide that they must flood the frame out all available ports. This is an unknown unicast condition and is their best effort to make sure that the frame eventually reaches its destination.

Step 3 Each switch floods or copies the frame to its 2/1 port on Segment B. PC-4, located on Segment B, receives the two frames destined for it. However, on Segment B, switch A now hears the new frame forwarded by switch B, and switch B hears the new frame forwarded by switch A.

Step 4 Switch A sees that the "new" frame is from PC-1 to PC-4. From the address table, the switch previously learned that PC-1 was on port 1/1, or Segment A. However, the source address of PC-1 has just been heard on port 2/1, or Segment B. By definition, the switch must relearn the location of PC-1 with the most recent information, which it now incorrectly assumes to be Segment B. (Switch B follows the same procedure, based on the "new" frame from switch A.)

Step 5 At this point, neither switch A nor switch B has learned the location of PC-4 because no frames have been received with PC-4 as the source address. Therefore, the new frame must be flooded out all available ports in an attempt to find PC-4. This frame then is sent out switch A's 1/1 port and onto Segment A, as well as switch B's 1/1 port and onto Segment A.

Step 6 Now both switches relearn the location of PC-1 as Segment A and forward the "new" frames back onto Segment B; then the entire process repeats.

This process of forwarding a single frame around and around between two switches is known as a *bridging loop*. Neither switch is aware of the other, so each happily forwards the same frame back and forth between its segments. Also note that because two switches are involved in the loop, the original frame has been duplicated and now is sent around in two counter-rotating loops. What stops the frame from being forwarded in this fashion forever? Nothing! PC-4 begins receiving frames addressed to it as fast as the switches can forward them.

Notice how the learned location of PC-1 keeps changing as frames get looped. Even a simple unicast frame has caused a bridging loop to form, and each switch's bridge table repeatedly is corrupted with incorrect data.

What would happen if PC-1 sent a broadcast frame instead? The bridging loops (remember that two of them are produced by the two parallel switches) form exactly as before. The broadcast frames continue to circulate forever. Now, however, every end-user device located on both Segments A and B receives and processes every broadcast frame. This type of broadcast storm easily can saturate the network segments and bring every host on the segments to a halt.

The only way to end the bridging loop condition is to physically break the loop by disconnecting switch ports or shutting down a switch. Obviously, it would be better to *prevent* bridging loops than to be faced with finding and breaking them after they form.

Preventing Loops with Spanning Tree Protocol

Bridging loops form because parallel switches (or bridges) are unaware of each other. STP was developed to overcome the possibility of bridging loops so that redundant switches and switch paths could be used for their benefits. Basically, the protocol enables switches to become aware of each other so they can negotiate a loop-free path through the network.

NOTE Because STP is involved in loop detection, many people refer to the catastrophic loops as "Spanning Tree loops." This is technically incorrect because the Spanning Tree Protocol's entire function is to *prevent* bridging loops. The correct terminology for this condition is a *bridging loop*.

Loops are discovered before they are made available for use, and redundant links effectively are shut down to prevent the loops from forming. In the case of redundant links, switches can be made aware that a link shut down for loop prevention should be brought up quickly in case of a link failure. The section "Redundant Link Convergence," in Chapter 10.

STP is communicated among all connected switches on a network. Each switch executes the Spanning Tree Algorithm based on information received from other neighboring switches. The algorithm chooses a reference point in the network and calculates all the redundant paths to that reference point. When redundant paths are found, the Spanning Tree Algorithm picks one path by which to forward frames and disables, or blocks, forwarding on the other redundant paths.

As its name implies, STP computes a tree structure that spans all switches in a subnet or network. Redundant paths are placed in a Blocking or Standby state to prevent frame forwarding. The switched network is then in a loop-free condition. However, if a forwarding port fails or becomes disconnected, the Spanning Tree Algorithm recomputes the spanning-tree topology so that the appropriate blocked links can be reactivated.

Spanning-Tree Communication: Bridge Protocol Data Units

STP operates as switches communicate with one another. Data messages are exchanged in the form of *Bridge Protocol Data Units (BPDU)*. A switch sends a BPDU frame out a port, using the unique MAC address of the port itself as a source address. The switch is unaware of the other switches around it, so BPDU frames are sent with a destination address of the well-known STP multicast address 01-80-c2-00-00-00.

Two types of BPDU exist:

- *Configuration BPDU*, used for spanning-tree computation
- *Topology Change Notification (TCN) BPDU*, used to announce changes in the network topology

The Configuration BPDU message contains the fields shown in Table 9-2. The TCN BPDU is discussed in the “Topology Changes” section later in this chapter.

Table 9-2 *Configuration BPDU Message Content*

Field Description	**Number of Bytes**
Protocol ID (always 0)	2
Version (always 0)	1
Message Type (Configuration or TCN BPDU)	1
Flags	1
Root Bridge ID	8
Root Path Cost	4
Sender Bridge ID	8
Port ID	2
Message Age (in 256ths of a second)	2
Maximum Age (in 256ths of a second)	2
Hello Time (in 256ths of a second)	2
Forward Delay (in 256ths of a second)	2

The exchange of BPDU messages works toward the goal of electing reference points as a foundation for a stable spanning-tree topology. Loops also can be identified and removed by placing specific redundant ports in a Blocking or Standby state. Notice that several key fields in the BPDU are related to bridge (or switch) identification, path costs, and timer values. These all work together so that the network of switches can converge upon a common spanning-tree topology and select the same reference points within the network. These reference points are defined in the sections that follow.

By default, BPDUs are sent out all switch ports every 2 seconds so that current topology information is exchanged and loops are identified quickly.

Electing a Root Bridge

For all switches in a network to agree on a loop-free topology, a common frame of reference must exist to use as a guide. This reference point is called the *Root Bridge.* (The term *bridge* continues to be used even in a switched environment because STP was developed for use in bridges. Therefore, when you see *bridge*, think *switch*.)

An election process among all connected switches chooses the Root Bridge. Each switch has a unique *Bridge ID* that identifies it to other switches. The Bridge ID is an 8-byte value consisting of the following fields:

- **Bridge Priority (2 bytes)**—The priority or weight of a switch in relation to all other switches. The priority field can have a value of 0 to 65,535 and defaults to 32,768 (or 0x8000) on every Catalyst switch.
- **MAC Address (6 bytes)**—The MAC address used by a switch can come from the Supervisor module, the backplane, or a pool of 1024 addresses that are assigned to every Supervisor or backplane, depending on the switch model. In any event, this address is hardcoded and unique, and the user cannot change it.

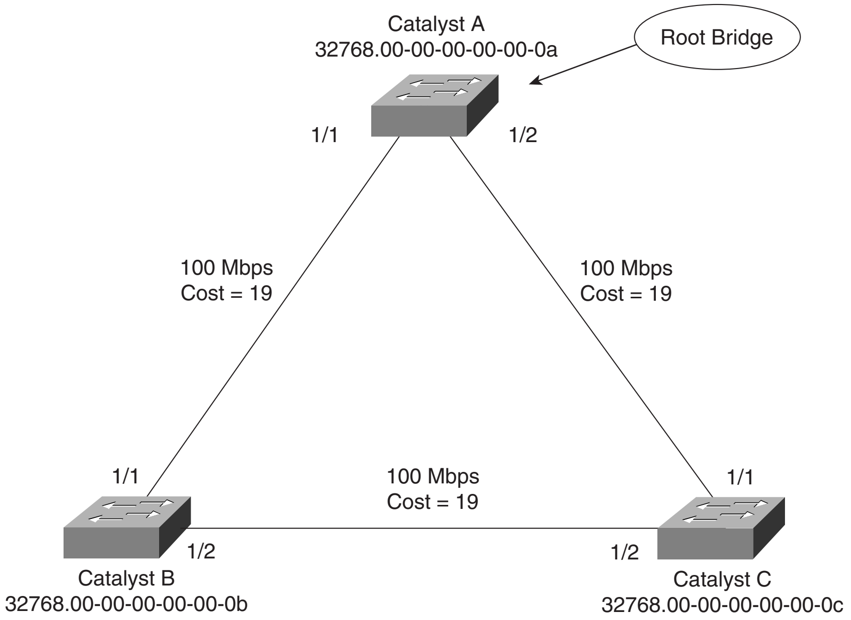
When a switch first powers up, it has a narrow view of its surroundings and assumes that it is the Root Bridge itself. (This notion probably will change as other switches check in and enter the election process.) The election process then proceeds as follows: Every switch begins by sending out BPDUs with a Root Bridge ID equal to its own Bridge ID and a Sender Bridge ID that is its own Bridge ID. The Sender Bridge ID simply tells other switches who is the actual sender of the BPDU message. (After a Root Bridge is decided upon, configuration BPDUs are sent only by the Root Bridge. All other bridges must forward or relay the BPDUs, adding their own Sender Bridge IDs to the message.)

Received BPDU messages are analyzed to see if a "better" Root Bridge is being announced. A Root Bridge is considered better if the Root Bridge ID value is *lower* than another. Again, think of the Root Bridge ID as being broken into Bridge Priority and MAC address fields. If two Bridge Priority values are equal, the lower MAC address makes the Bridge ID better. When a switch hears of a better Root Bridge, it replaces its own Root Bridge ID with the Root Bridge ID announced in the BPDU. The switch then is required to recommend or advertise the new Root Bridge ID in its own BPDU messages, although it still identifies itself as the Sender Bridge ID.

Sooner or later, the election converges and all switches agree on the notion that one of them is the Root Bridge. As might be expected, if a new switch with a lower Bridge Priority powers up, it begins advertising itself as the Root Bridge. Because the new switch does indeed have a lower Bridge ID, all the switches soon reconsider and record it as the new Root Bridge. This also can happen if the new switch has a Bridge Priority equal to that of the existing Root Bridge but has a lower MAC address. Root Bridge election is an ongoing process, triggered by Root Bridge ID changes in the BPDUs every 2 seconds.

As an example, consider the small network shown in Figure 9-3. For simplicity, assume that each Catalyst switch has a MAC address of all 0s, with the last hex digit equal to the switch label.

Figure 9-3 *Example of Root Bridge Election*



In this network, each switch has the default Bridge Priority of 32,768. The switches are interconnected with Fast Ethernet links. All three switches try to elect themselves as the Root, but all of them have equal Bridge Priority values. The election outcome produces the Root Bridge, determined by the lowest MAC address—that of Catalyst A.

Electing Root Ports

Now that a reference point has been nominated and elected for the entire switched network, each nonroot switch must figure out where it is in relation to the Root Bridge. This action can be performed by selecting only one *Root Port* on each nonroot switch. The Root Port always points toward the current Root Bridge.

STP uses the concept of cost to determine many things. Selecting a Root Port involves evaluating the *Root Path Cost*. This value is the cumulative cost of all the links leading to the Root Bridge. A particular switch link also has a cost associated with it, called the *Path Cost*. To understand the difference between these values, remember that only the Root Path Cost is carried inside the BPDU. (See Table 9-2 again.) As the Root Path Cost travels along, other switches can modify its value to make it cumulative. The Path Cost, however, is not contained in the BPDU. It is known only to the local switch where the port (or "path" to a neighboring switch) resides.

Path Costs are defined as a 1-byte value, with the default values shown in Table 9-3. Generally, the higher the bandwidth of a link, the lower the cost of transporting data across it. The original IEEE 802.1D standard defined Path Cost as 1000 Mbps divided by the link bandwidth in megabits per second. These values are shown in the center column of the table. Modern networks commonly use Gigabit Ethernet and OC-48 ATM, which are both either too close to or greater than the maximum scale of 1000 Mbps. The IEEE now uses a nonlinear scale for Path Cost, as shown in the right column of the table.

TIP Be aware that there are two STP path cost scales, one that is little used with a linear scale and one commonly used that is nonlinear. If you decide to memorize some common Path Cost values, learn only the ones in the "new" right column of the table.

Table 9-3 *STP Path Cost*

Link Bandwidth	Old STP Cost	New STP Cost
4 Mbps	250	250
10 Mbps	100	100
16 Mbps	63	62
45 Mbps	22	39
100 Mbps	10	19
155 Mbps	6	14
622 Mbps	2	6
1 Gbps	1	4
10 Gbps	0	2

The Root Path Cost value is determined in the following manner:

1. The Root Bridge sends out a BPDU with a Root Path Cost value of 0 because its ports sit directly on the Root Bridge.
2. When the next-closest neighbor receives the BPDU, it adds the Path Cost of its own port where the BPDU arrived. (This is done as the BPDU is *received*.)
3. The neighbor sends out BPDUs with this new cumulative value as the Root Path Cost.

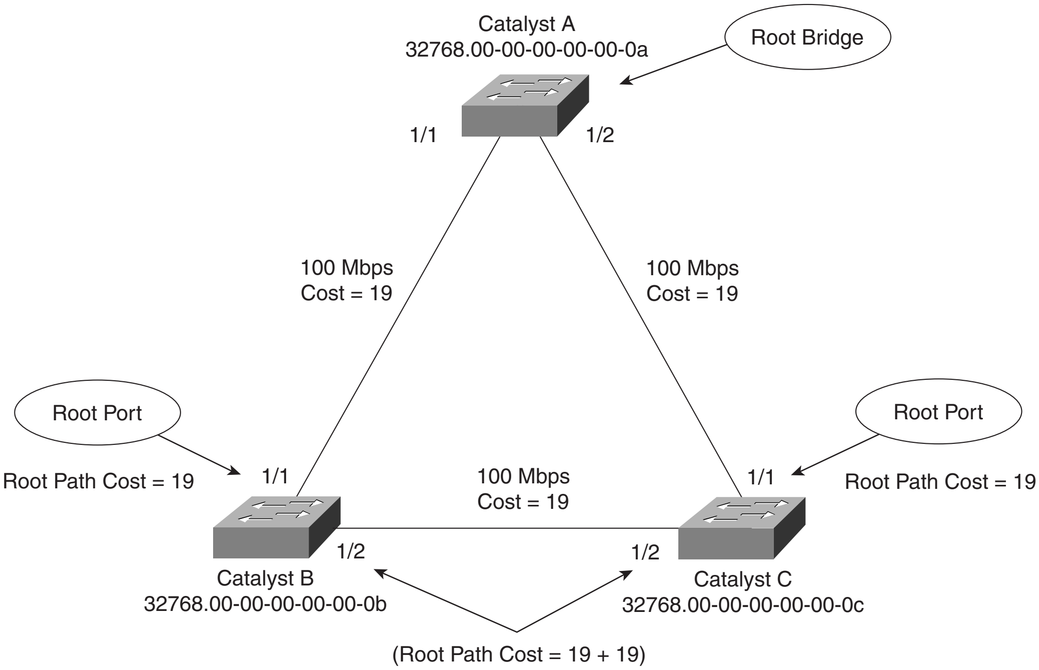
4. The Root Path Cost is incremented by the ingress port Path Cost as the BPDU is received at each switch down the line.

5. Notice the emphasis on incrementing the Root Path Cost as BPDUs are *received*. When computing the Spanning Tree Algorithm manually, remember to compute a new Root Path Cost as BPDUs *come in* to a switch port, not as they go out.

After incrementing the Root Path Cost, a switch also records the value in its memory. When a BPDU is received on another port and the new Root Path Cost is lower than the previously recorded value, this lower value becomes the new Root Path Cost. In addition, the lower cost tells the switch that the path to the Root Bridge must be better using this port than it was on other ports. The switch now has determined which of its ports has the best path to the Root: the Root Port.

Figure 9-4 shows the same network from Figure 9-3 in the process of Root Port selection.

Figure 9-4 *Example of Root Port Selection*



The Root Bridge, Catalyst A, already has been elected. Therefore, every other switch in the network must choose one port that has the best path to the Root Bridge. Catalyst B selects its port 1/1, with a Root Path Cost of 0 plus 19. Port 1/2 is not chosen because its Root Path Cost is 0 (BPDU from Catalyst A) plus 19 (Path Cost of A–C link), plus 19 (Path Cost of C–B link), or a total of 38. Catalyst C makes a similar choice of port 1/1.

Electing Designated Ports

By now, you should begin to see the process unfolding: A starting or reference point has been identified, and each switch “connects” itself toward the reference point with the single link that has the best path. A tree structure is beginning to emerge, but links have only been identified at this point. All links still are connected and could be active, leaving bridging loops.

To remove the possibility of bridging loops, STP makes a final computation to identify one *Designated Port* on each network segment. Suppose that two or more switches have ports connected to a single common network segment. If a frame appears on that segment, all the bridges attempt to forward it to its destination. Recall that this behavior was the basis of a bridging loop and should be avoided.

Instead, only one of the links on a segment should forward traffic to and from that segment—the one that is selected as the Designated Port. Switches choose a Designated Port based on the lowest cumulative Root Path Cost to the Root Bridge. For example, a switch always has an idea of its own Root Path Cost, which it announces in its own BPDUs. If a neighboring switch on a shared LAN segment sends a BPDU announcing a lower Root Path Cost, the neighbor must have the Designated Port. If a switch learns only of higher Root Path Costs from other BPDUs received on a port, however, it then correctly assumes that its own receiving port is the Designated Port for the segment.

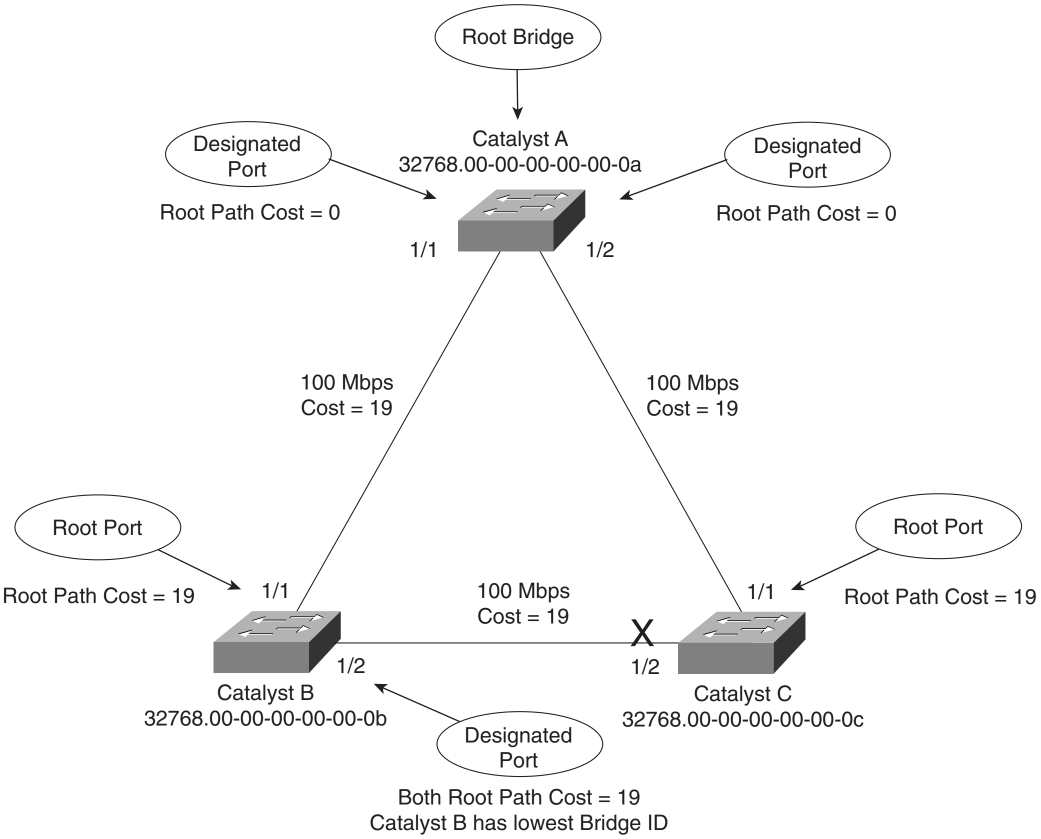
Notice that the entire STP determination process has served only to identify bridges and ports. All ports are still active, and bridging loops still might lurk in the network. STP has a set of progressive states that each port must go through, regardless of the type or identification. These states actively prevent loops from forming and are described in the next section.

NOTE In each determination process discussed so far, two or more links might have identical Root Path Costs. This results in a tie condition, unless other factors are considered. All tie-breaking STP decisions are based on the following sequence of four conditions:

1. Lowest Root Bridge ID
2. Lowest Root Path Cost to Root Bridge
3. Lowest Sender Bridge ID
4. Lowest Sender Port ID

Figure 9-5 demonstrates an example of Designated Port selection. This figure is identical to Figure 9-3 and Figure 9-4, with further spanning-tree development shown. The only changes are the choices of Designated Ports, although seeing all STP decisions shown on one network diagram is handy.

Figure 9-5 *Example of Designated Port Selection*



The three switches have chosen their Designated Ports (DP) for the following reasons:

- **Catalyst A**—Because this switch is the Root Bridge, all its active ports are Designated Ports, by definition. At the Root Bridge, the Root Path Cost of each port is 0.

- **Catalyst B**—Catalyst A port 1/1 is the DP for the Segment A–B because it has the lowest Root Path Cost (0). Catalyst B port 1/2 is the DP for segment B–C. The Root Path Cost for each end of this segment is 19, determined from the incoming BPDU on port 1/1. Because the Root Path Cost is equal on both ports of the segment, the DP must be chosen by the next criteria—the lowest Sender Bridge ID. When Catalyst B sends a BPDU to Catalyst C, it has the lowest MAC address in the Bridge ID. Catalyst C also sends a BPDU to Catalyst B, but its Sender Bridge ID is higher. Therefore, Catalyst B port 1/2 is selected as the segment's DP.

- **Catalyst C**—Catalyst A port 1/2 is the DP for Segment A–C because it has the lowest Root Path Cost (0). Catalyst B port 1/2 is the DP for Segment B–C. Therefore, Catalyst C port 1/2 will be neither a Root Port nor a Designated Port. As discussed in the next section, any port that is not elected to either position enters the Blocking state. Where blocking occurs, bridging loops are broken.

STP States

To participate in STP, each port of a switch must progress through several states. A port begins its life in a Disabled state, moving through several passive states and, finally, into an active state if allowed to forward traffic. The STP port states are as follows:

- **Disabled**—Ports that are administratively shut down by the network administrator, or by the system because of a fault condition, are in the Disabled state. This state is special and is not part of the normal STP progression for a port.
- **Blocking**—After a port initializes, it begins in the Blocking state so that no bridging loops can form. In the Blocking state, a port cannot receive or transmit data and cannot add MAC addresses to its address table. Instead, a port is allowed to receive only BPDUs so that the switch can hear from other neighboring switches. In addition, ports that are put into standby mode to remove a bridging loop enter the Blocking state.
- **Listening**—A port is moved from Blocking to Listening if the switch thinks that the port can be selected as a Root Port or Designated Port. In other words, the port is on its way to begin forwarding traffic.

  In the Listening state, the port still cannot send or receive data frames. However, the port is allowed to receive and send BPDUs so that it actively can participate in the Spanning Tree topology process. Here, the port finally is allowed to become a Root Port or Designated Port because the switch can advertise the port by sending BPDUs to other switches. If the port loses its Root Port or Designated Port status, it returns to the Blocking state.
- **Learning**—After a period of time called the Forward Delay in the Listening state, the port is allowed to move into the Learning state. The port still sends and receives BPDUs as before. In addition, the switch now can learn new MAC addresses to add to its address table. This gives the port an extra period of silent participation and allows the switch to assemble at least some address table information. The port cannot yet send any data frames, however.
- **Forwarding**—After another Forward Delay period of time in the Learning state, the port is allowed to move into the Forwarding state. The port now can send and receive data frames, collect MAC addresses in its address table, and send and receive BPDUs. The port is now a fully functioning switch port within the spanning-tree topology.

> Remember that a switch port is allowed into the Forwarding state only if no redundant links (or loops) are detected and if the port has the best path to the Root Bridge as the Root Port or Designated Port.

Example 9-1 shows the output from a switch as one of its ports progresses through the STP port states.

Example 9-1 *Port Progressing Through the STP Port States*

```
*Mar 16 14:31:00 UTC: STP SW: Fa0/1 new disabled req for 1 vlans
Switch(config)# interface fast 0/1
Switch(config-if)#no shut
Switch(config-if)#^-Z
*Mar 16 14:31:00 UTC: STP SW: Fa0/1 new blocking req for 1 vlans

Switch#show spanning interface fast 0/1

Vlan             Port ID                     Designated                Port ID
Name             Prio.Nbr      Cost Sts      Cost Bridge ID            Prio.Nbr
---------------- -------- --------- --- --------- -------------------- --------
VLAN0001         128.1           19 LIS         0 32769 000a.f40a.2980 128.1

*Mar 16 14:31:15 UTC: STP SW: Fa0/1 new learning req for 1 vlans

Switch#show spanning interface fast 0/1
Vlan             Port ID                     Designated                Port ID
Name             Prio.Nbr      Cost Sts      Cost Bridge ID            Prio.Nbr
---------------- -------- --------- --- --------- -------------------- --------
VLAN0001         128.1           19 LRN         0 32768 00d0.5849.4100  32.129

*Mar 16 14:31:30 UTC: STP SW: Fa0/1 new forwarding req for 1 vlans

Switch#show spanning interface fast 0/1

Vlan             Port ID                     Designated                Port ID
Name             Prio.Nbr      Cost Sts      Cost Bridge ID            Prio.Nbr
---------------- -------- --------- --- --------- -------------------- --------
VLAN0001         128.1           19 FWD         0 32768 00d0.5849.4100  32.129
```

The example begins as the port administratively is disabled from the command line. When the port is enabled, successive **show spanning-tree interface** *type mod/port* commands display the port state as Listening, Learning, and then Forwarding. These are shown in the shaded text of the example. Notice also the time stamps and port states provided by the **debug spanning-tree switch state** command, which give a sense of the timing between port states. Because this port was eligible as a Root Port, the **show** command never could execute fast enough to show the port in the Blocking state.

STP Timers

STP operates as switches send BPDUs to each other in an effort to form a loop-free topology. The BPDUs take a finite amount of time to travel from switch to switch. In addition, news of a topology change (such as a link or Root Bridge failure) can suffer from propagation delays as the announcement travels from one side of a network to the other. Because of the possibility of these delays, keeping the spanning-tree topology from settling out or converging until all switches have had time to receive accurate information is important.

STP uses three timers to make sure that a network converges properly before a bridging loop can form. The timers and their default values are as follows:

- **Hello Time**—The time interval between Configuration BPDUs sent by the Root Bridge. The Hello Time value configured in the Root Bridge switch determines the Hello Time for all nonroot switches because they just relay the Configuration BPDUs as they are received from the root. However, all switches have a locally configured Hello Time that is used to time TCN BPDUs when they are retransmitted. The IEEE 802.1D standard specifies a default Hello Time value of 2 seconds.
- **Forward Delay**—The time interval that a switch port spends in both the Listening and Learning states. The default value is 15 seconds.
- **Max (maximum) Age**—The time interval that a switch stores a BPDU before discarding it. While executing the STP, each switch port keeps a copy of the "best" BPDU that it has heard. If the switch port loses contact with the BPDU's source (no more BPDUs are received from it), the switch assumes that a topology change must have occurred after the Max Age time elapsed and so the BPDU is aged out. The default Max Age value is 20 seconds.

The STP timers can be configured or adjusted from the switch command line. However, the timer values never should be changed from the defaults without careful consideration. Then the values should be changed only on the Root Bridge switch. Recall that the timer values are advertised in fields within the BPDU. The Root Bridge ensures that the timer values propagate to all other switches.

TIP The default STP timer values are based on some assumptions about the size of the network and the length of the Hello Time. A reference model of a network having a diameter of seven switches derives these values. The diameter is measured from the Root Bridge switch outward, including the Root Bridge.

In other words, if you drew the STP topology, the diameter would be the number of switches connected in series from the Root Bridge out to the end of any branch in the tree. The Hello Time is based on the time it takes for a BPDU to travel from the Root Bridge to a point seven switches away. This computation uses a Hello Time of 2 seconds.

The network diameter can be configured on the Root Bridge switch to more accurately reflect the true size of the physical network. Making that value more accurate reduces the total STP convergence time during a topology change. Cisco also recommends that if changes need to be made, only the network diameter value should be modified on the Root Bridge switch. When the diameter is changed, the switch calculates new values for all three timers automatically.

Topology Changes

To announce a change in the active network topology, switches send a TCN BPDU. Table 9-4 shows the format of these messages.

Table 9-4 *Topology Change Notification BPDU Message Content*

Field Description	# of Bytes
Protocol ID (always 0)	2
Version (always 0)	1
Message Type (Configuration or TCN BPDU)	1

A topology change occurs when a switch either moves a port into the Forwarding state or moves a port from the Forwarding or Learning states into the Blocking state. In other words, a port on an active switch comes up or goes down. The switch sends a TCN BPDU out its Root Port so that, ultimately, the Root Bridge receives news of the topology change. Notice that the TCN BPDU carries no data about the change but informs recipients only that a change has occurred. Also notice that the switch will not send TCN BPDUs if the port has been configured with PortFast enabled.

The switch continues sending TCN BPDUs every Hello Time interval until it gets an acknowledgment from its upstream neighbor. As the upstream neighbors receive the TCN BPDU, they propagate it on toward the Root Bridge and send their own acknowledgments. When the Root Bridge receives the TCN BPDU, it also sends out an acknowledgment. However, the Root Bridge sets the Topology Change flag in its Configuration BPDU, which is relayed to every other bridge in the network. This is done to signal the topology change and cause all other bridges to shorten their bridge table aging times from the default (300 seconds) to only the Forward Delay value (default 15 seconds).

This condition causes the learned locations of MAC addresses to be flushed out much sooner than they normally would, easing the bridge table corruption that might occur because of the change in topology. However, any stations that actively are communicating during this time are kept in the bridge table. This condition lasts for the sum of the Forward Delay and the Max Age (default 15 + 20 seconds).

The theory behind topology changes is fairly straightforward, but it’s often difficult to grasp how a working network behaves during a change. For example, suppose you have a Layer 2 network (think of a single VLAN or a single instance of STP) that is stable and loop free. If a switch uplink suddenly failed or a new uplink was added, how would the various switches in the network react? Would users all over the network lose connectivity while the STP "recomputes" or reconverges?

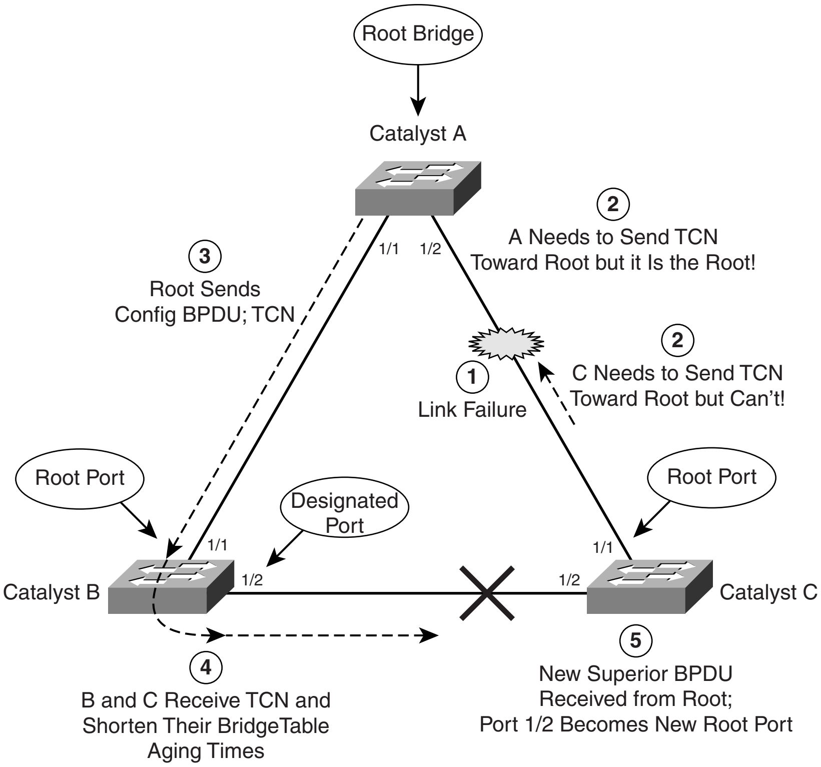
Examples of different types of topology changes are presented in the following sections, along with the sequence of STP events. Each type has a different cause and a different effect. To provide continuity as the STP concepts are presented, the same network previously shown in Figures 9-3 through 9-5 is used in each of these examples.

Direct Topology Changes

A direct topology change is one that can be detected on a switch interface. For example, if a trunk link suddenly goes down, the switch on each end of the link immediately can detect a link failure. The absence of that link changes the bridging topology, so other switches should be notified.

Figure 9-6 shows a network that has converged into a stable STP topology. The VLAN is forwarding on all trunk links except port 1/2 on Catalyst C, where it is in the Blocking state.

Figure 9-6 *Effects of a Direct Topology Change*



This network has just suffered a link failure between Catalyst A and Catalyst C. The sequence of events unfolds as follows:

1. Catalyst C detects a link down on its port 1/1; Catalyst A detects a link down on its port 1/2.
2. Catalyst C removes the previous “best” BPDU it had received from the Root over port 1/1. Port 1/1 is now down so that BPDU is no longer valid.

   Normally, Catalyst C would try to send a TCN message out its Root Port, to reach the Root Bridge. Here, the Root Port is broken, so that isn’t possible. Without an advanced feature such as STP UplinkFast, Catalyst C isn’t yet aware that another path exists to the Root.

   Also, Catalyst A is aware of the link down condition on its own port 1/2. It normally would try to send a TCN message out its Root Port, to reach the Root Bridge. Here, Catalyst A *is* the Root, so that isn’t really necessary.
3. The Root Bridge, Catalyst A, sends a Configuration BPDU with the TCN bit set out its port 1/1. This is received and relayed by each switch along the way, informing each one of the topology change.
4. Catalysts B and C receive the TCN message. The only reaction these switches take is to shorten their bridging table aging times to the Forward Delay time. At this point, they don’t know how the topology has changed; they only know to force fairly recent bridging table entries to age out.
5. Catalyst C basically just sits and waits to hear from the Root Bridge again. The Config BPDU TCN message is received on port 1/2, which was previously in the Blocking state. This BPDU becomes the “best” one received from the Root, so port 1/2 becomes the new Root Port.

   Catalyst C now can progress port 1/2 from Blocking through the Listening, Learning, and Forwarding states.

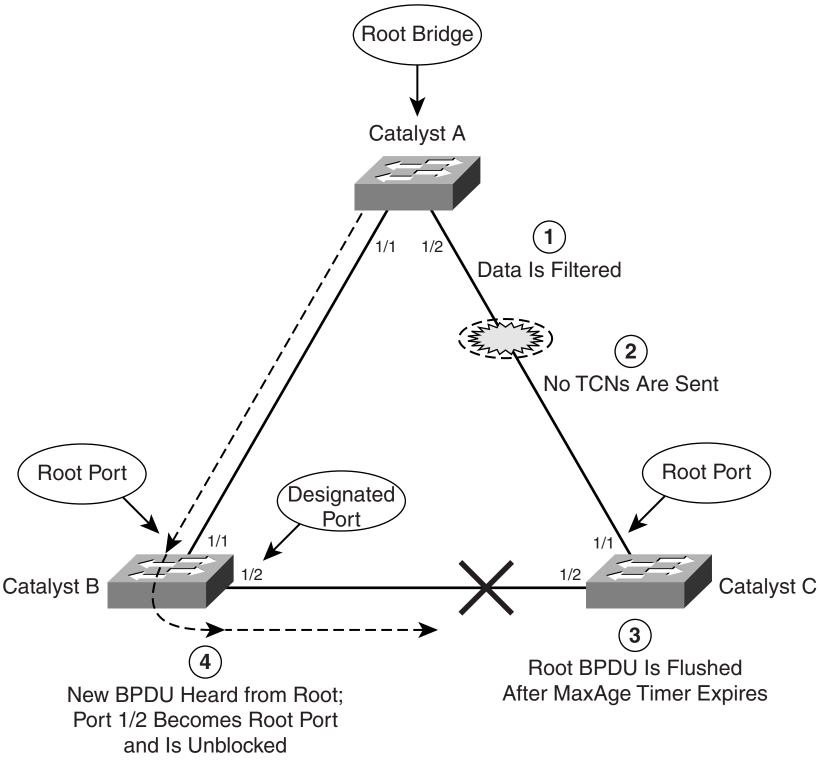
As a result of a direct link failure, the topology has changed and STP has converged again. Notice that only Catalyst C has undergone any real effects from the failure. Switches A and B heard the news of the topology change but did not have to move any links through the STP states. In other words, the whole network did not go through a massive STP reconvergence.

The total time that users on Catalyst C lost connectivity was roughly the time that port 1/2 spent in the Listening and Learning states. With the default STP timers, this amounts to about two times the Forward Delay period (15 seconds), or 30 seconds total.

Indirect Topology Changes

Figure 9-7 shows the same network as Figure 9-6, but this time, the link failure indirectly involves Catalysts A and C. The link status at each switch stays up, but something between them has failed or is filtering traffic. This could be another device, such as a service provider's switch, a firewall, and so on. As a result, no data (including BPDUs) can pass between those switches.

Figure 9-7 *Effects of a Indirect Topology Change*



STP can detect and recover from indirect failures, thanks to timer mechanisms. The sequence of events unfolds as follows:

1. Catalysts A and C both show a link up condition; data begins to be filtered elsewhere on the link.
2. No link failure is detected, so no TCN messages are sent.
3. Catalyst C already has stored the "best" BPDU it had received from the Root over port 1/1. No further BPDUs are received from the Root over that port. After the MaxAge timer expires, no other BPDU is available to refresh the "best" entry, so it is flushed. Catalyst C now must wait to hear from the Root again on any of its ports.

4. The next Configuration BPDU from the Root is heard on Catalyst C port 1/2. This BPDU becomes the new "best" entry, and port 1/2 becomes the Root Port. Now the port is progressed from Blocking through the Listening, Learning, and finally Forwarding states.

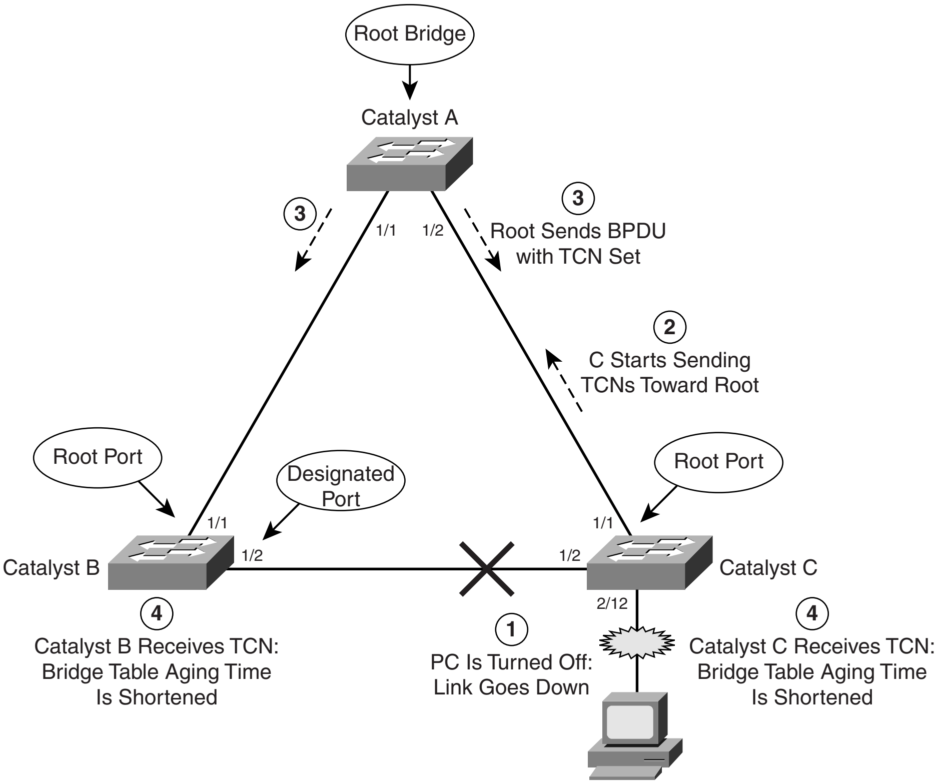
As a result of the indirect link failure, the topology doesn't change immediately. The absence of BPDUs from the Root causes Catalyst C to take some action. Because this type of failure relies on STP timer activity, it generally takes longer to detect and mitigate.

In this example, the total time that users on Catalyst C lost connectivity was roughly the time until the MaxAge timer expired (20 seconds), plus the time until the next Configuration BPDU was received (2 seconds) on port 1/2, plus the time that port 1/2 spent in the Listening (15 seconds) and Learning (15 seconds) states. In other words, 52 seconds elapse if the default timer values are used.

Insignificant Topology Changes

Figure 9-8 shows the same network topology as Figure 9-6 and Figure 9-7, with the addition of a user PC on access-layer switch Catalyst C. The user's switch port, 2/12, is just another link as far as the switch is concerned. If the link status goes up or down, the switch must view that as a topology change and inform the Root Bridge.

Figure 9-8 *Effects of an Insignificant Topology Change*


Root Bridge
Catalyst A
3
1/1
1/2
3
Root Sends BPDU with TCN Set
2
C Starts Sending TCNs Toward Root
Root Port
Designated Port
Root Port
1/1
1/2
1/1
1/2
Catalyst B
Catalyst C
2/12
4
Catalyst B Receives TCN: Bridge Table Aging Time Is Shortened
1
PC Is Turned Off: Link Goes Down
4
Catalyst C Receives TCN: Bridge Table Aging Time Is Shortened

Obviously, user ports are expected to go up and down as the users reboot their machines, turn them on and off as they go to and from work, and so on. Regardless, TCN messages are sent by the switch, just as if a trunk link between switches had changed state.

To see what effect this has on the STP topology and the network, consider the following sequence of events:

1. The PC on Catalyst port 2/12 is turned off. The switch detects the link status going down.
2. Catalyst C begins sending TCN BPDUs toward the Root, over its Root Port (1/1).
3. The Root sends a TCN acknowledgment back to Catalyst C and then sends a Configuration BPDU with the TCN bit set to all downstream switches. This is done to inform every switch of a topology change somewhere in the network.
4. The TCN flag is received from the Root, and both Catalysts B and C shorten their bridge table aging times. This causes recently idle entries to be flushed, leaving only the actively transmitting stations in the table. The aging time stays short for the duration of the Forward Delay and Max Age timers.

Notice that this type of topology change is mostly cosmetic. No actual topology change occurred because none of the switches had to change port states to reach the Root Bridge. Instead, powering off the PC caused all the switches to age out entries from their bridge or CAM tables much sooner than normal.

At first, this doesn't seem like a major problem because the PC link state affects only the "newness" of the CAM table contents. If CAM table entries are flushed as a result, they probably will be learned again. This becomes a problem when every user PC is considered. Now every time *any* PC in the network powers up or down, *every* switch in the network must age out CAM table entries.

Given enough PCs, the switches could be in a constant state of flushing bridge tables. Also remember that when a switch doesn't have a CAM entry for a destination, the packet must be flooded out all its ports. Flushed tables mean more unknown unicasts, which mean more broadcasts or flooded packets throughout the network.

Fortunately, Catalyst switches have a feature that can designate a port as a special case. You can enable the STP PortFast feature on a port with a single attached PC. As a result, TCNs aren't sent when the port changes state, and the port is brought right into the Forwarding state when the link comes up. The section "Redundant Link Convergence," in Chapter 10, covers PortFast in more detail.

Types of STP

So far, this chapter has discussed STP in terms of its operation to prevent loops and to recover from topology changes in a timely manner. STP originally was developed to operate in a bridged environment, basically supporting a single LAN (or one VLAN). Implementing STP into a switched environment has required additional consideration and modification to support multiple VLANs. Because of this, the IEEE and Cisco have approached STP differently. This section reviews the three traditional types of STP that are encountered in switched networks and how they relate to one another. No specific configuration commands are associated with the various types of STP here. Instead, you need a basic understanding of how they interoperate in a network.

NOTE The IEEE has produced additional standards for spanning-tree enhancements that greatly improve on its scalability and convergence aspects. These are covered in Chapter 12. When you have a firm understanding of the more traditional forms of STP presented in this chapter, you can grasp the enhanced versions much easier.

Common Spanning Tree

The IEEE 802.1Q standard specifies how VLANs are to be trunked between switches. It also specifies only a single instance of STP that encompasses all VLANs. This instance is referred to as the *Common Spanning Tree (CST).* All CST BPDUs are transmitted over trunk links using the native VLAN with untagged frames.

Having a single STP for many VLANs simplifies switch configuration and reduces switch CPU load during STP calculations. However, having only one STP instance can cause limitations, too. Redundant links between switches will be blocked with no capability for load balancing. Conditions also can occur that would cause CST to mistakenly enable forwarding on a link that does not carry a specific VLAN, while other links would be blocked.

Per-VLAN Spanning Tree

Cisco has a proprietary version of STP that offers more flexibility than the CST version. *Per-VLAN Spanning Tree* (*PVST)* operates a separate instance of STP for each individual VLAN. This allows the STP on each VLAN to be configured independently, offering better performance and tuning for specific conditions. Multiple spanning-trees also make load balancing possible over redundant links when the links are assigned to different VLANs. One link might forward one set of VLANs, while another redundant link might forward a different set.

Because of its proprietary nature, PVST requires the use of Cisco Inter-Switch Link (ISL) trunking encapsulation between switches. In networks where PVST and CST coexist, interoperability problems occur. Each requires a different trunking method, so BPDUs are never exchanged between STP types.

Per-VLAN Spanning Tree Plus

Cisco has a second proprietary version of STP that allows devices to interoperate with both PVST and CST. *Per-VLAN Spanning Tree Plus (PVST+)* effectively supports three groups of STP operating in the same campus network:

- Catalyst switches running PVST
- Catalyst switches running PVST+
- Switches running CST over 802.1Q

To do this, PVST+ acts as a translator between groups of CST switches and groups of PVST switches. PVST+ can communicate directly with PVST by using ISL trunks. To communicate with CST, however, PVST+ exchanges BPDUs with CST as untagged frames over the native VLAN. BPDUs from other instances of STP (other VLANs) are propagated across the CST portions of the network by tunneling. PVST+ sends these BPDUs by using a unique multicast address so that the CST switches forward them on to downstream neighbors without interpreting them first. Eventually, the tunneled BPDUs reach other PVST+ switches where they are understood.

Foundation Summary

The Foundation Summary is a collection of information that provides a convenient review of many key concepts in this chapter. If you are already comfortable with the topics in this chapter, this summary can help you recall a few details. If you just read this chapter, this review should help solidify some key facts. If you are doing your final preparation before the exam, this information is a convenient way to review the day before the exam.

STP has a progression of states that each port moves through. Each state allows a port to do only certain functions, as shown in Table 9-5.

Table 9-5 *STP States and Port Activity*

STP State	The Port Can…	The Port Cannot…	Duration
Disabled		Send or receive data	
Blocking	Receive BPDUs	Send or receive data or learn MAC addresses	Indefinite if loop has been detected
Listening	Send and receive BPDUs	Send or receive data or learn MAC addresses	Forward Delay timer (15 seconds)
Learning	Send and receive BPDUs and learn MAC addresses	Send or receive data	Forward Delay timer (15 seconds)
Forwarding	Send and receive BPDUs, learn MAC addresses, and send and receive data		Indefinite as long as port is up and loop is not detected

Table 9-6 *Basic Spanning-Tree Operation*

Task	Procedure
1. Elect Root Bridge.	Lowest Bridge ID
2. Select Root Port (one per switch).	Lowest Root Path Cost; if equal, use tie-breakers
3. Select Designated Port (one per segment).	Lowest Root Path Cost; if equal, use tie-breakers
4. Block ports with loops.	Block ports that are non-Root and non–Designated Ports

To manually work out a spanning-tree topology using a network diagram, follow the basic steps in Table 9-7.

Table 9-7 *Manual STP Computation*

Task	Description
1. Identify Path Costs on links.	For each link between switches, write the Path Cost that each switch uses for the link.
2. Identify Root Bridge.	Find the switch with the lowest Bridge ID; mark it on the drawing.
3. Select Root Ports (one per switch).	For each switch, find the one port that has the best path to the Root Bridge. This is the one with the lowest Root Path Cost. Mark the port with an RP label.
4. Select Designated Ports (one per segment).	For each link between switches, identify which end of the link will be the Designated Port. This is the one with the lowest Root Path Cost; if equal on both ends, use STP tie-breakers. Mark the port with a DP label.
5. Identify the blocking ports.	Every switch port that is neither a Root nor a Designated Port will be put into the Blocking state. Mark these with an X.

Table 9-8 *Spanning-Tree Tie-Breaker Criteria*

Sequence	Criteria
1	Lowest Root Bridge ID
2	Lowest Root Path Cost
3	Lowest Sender Bridge ID
4	Lowest Sender Port ID

Table 9-9 *STP Path Cost*

Link Bandwidth	STP Cost (Nonlinear Scale)
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14

continues

Table 9-9 *STP Path Cost (Continued)*

622 Mbps	6
1 Gbps	4
10 Gbps	2

Table 9-10 *STP Timers*

Timer	Function	Default Value
Hello	Interval between Configuration BPDUs.	2 seconds
Forward Delay	Time spent in Listening and Learning states before transitioning toward Forwarding state.	15 seconds
Max Age	Maximum length of time a BPDU can be stored without receiving an update. Timer expiration signals an indirect failure with Designated or Root Bridge.	20 seconds

Table 9-11 *Types of STP*

Type of STP	Function
CST	One instance of STP, over the native VLAN; 802.1Q based
PVST	One instance of STP per VLAN; Cisco ISL based
PVST+	Provides interoperability between CST and PVST; operates over both 802.1Q and ISL

Q&A

The questions and scenarios in this book are more difficult than what you should experience on the actual exam. The questions do not attempt to cover more breadth or depth than the exam; however, they are designed to make sure that you know the answers. Rather than allowing you to derive the answers from clues hidden inside the questions themselves, the questions challenge your understanding and recall of the subject. Hopefully, these questions will help limit the number of exam questions on which you narrow your choices to two options and then guess.

You can find the answers to these questions in Appendix A.

1. What is a bridging loop? Why is it bad?
2. Put the following STP port states in chronological order:
   a. Learning
   b. Forwarding
   c. Listening
   d. Blocking
3. Choose two types of STP messages used to communicate between bridges:
   a. Advertisement BPDU
   b. Configuration BPDU
   c. ACK BPDU
   d. TCN BPDU
4. What criteria are used to select the following?
   a. Root Bridge
   b. Root Port
   c. Designated Port
   d. Redundant (or secondary) Root Bridges
5. Which of the following switches becomes the Root Bridge, given the information in the following table? Which switch becomes the secondary Root Bridge if the Root Bridge fails?

Switch Name	Bridge Priority	MAC Address	Port Costs
Catalyst A	32,768	00-d0-10-34-26-a0	All are 19
Catalyst B	32,768	00-d0-10-34-24-a0	All are 4
Catalyst C	32,767	00-d0-10-34-27-a0	All are 19
Catalyst D	32,769	00-d0-10-34-24-a1	All are 19

6. What conditions cause an STP topology change? What effect does this have on STP and the network?
7. A Root Bridge has been elected in a switched network. Suppose that a new switch is installed with a lower Bridge ID than the existing Root Bridge. What will happen?
8. Suppose that a switch receives Configuration BPDUs on two of its ports. Both ports are assigned to the same VLAN. Each of the BPDUs announces Catalyst A as the Root Bridge. Can the switch use both of these ports as Root Ports? Why?
9. How is the Root Path Cost calculated for a switch port?
10. What conditions can cause ports on a network's Root Bridge to move into the Blocking state? (Assume that all switch connections are to other switches. No crossover cables are used to connect two ports on the same switch.)
11. What parameters can be tuned to influence the selection of a port as a Root or Designated Port?
12. After a bridging loop forms, how can you stop the endless flow of traffic?
13. In a BPDU, when can the Root Bridge ID have the same value as the Sender Bridge ID?
14. Which of these is true about the Root Path Cost?
    a. It is a value sent by the Root Bridge that cannot be changed along the way.
    b. It is incremented as a switch receives a BPDU.
    c. It is incremented as a switch sends a BPDU.
    d. It is incremented by the Path Cost of a port.
15. Suppose that two switches are connected by a common link. Each must decide which one will have the Designated Port on the link. Which switch takes on this role if these STP advertisements occur?
    a. The link is on switch A's port number 12 and on switch B's port number 5.
    b. Switch A has a Bridge ID of 32,768:0000.1111.2222, and switch B has 8192:0000.5555.6666.
    c. Switch A advertises a Root Path Cost of 8, while B advertises 12.
16. Using the default STP timers, how long does it take for a port to move from the Blocking state to the Forwarding state?
17. If the Root Bridge sets the Topology Change flag in the BPDU, what must the other switches in the network do?

18. Over what VLANs does the CST form of STP run?

a. VLAN 1

b. All active VLANs

c. All VLANs (active or inactive)

d. The native VLAN

19. What is the major difference between PVST and PVST+?

20. Two switches are connected by a common active link. When might neither switch have a Designated Port on the link?

a. When neither has a better Root Path Cost.

b. When the switches are actually the primary and secondary Root Bridges.

c. When one switch has its port in the Blocking state.

d. Never; this can't happen.