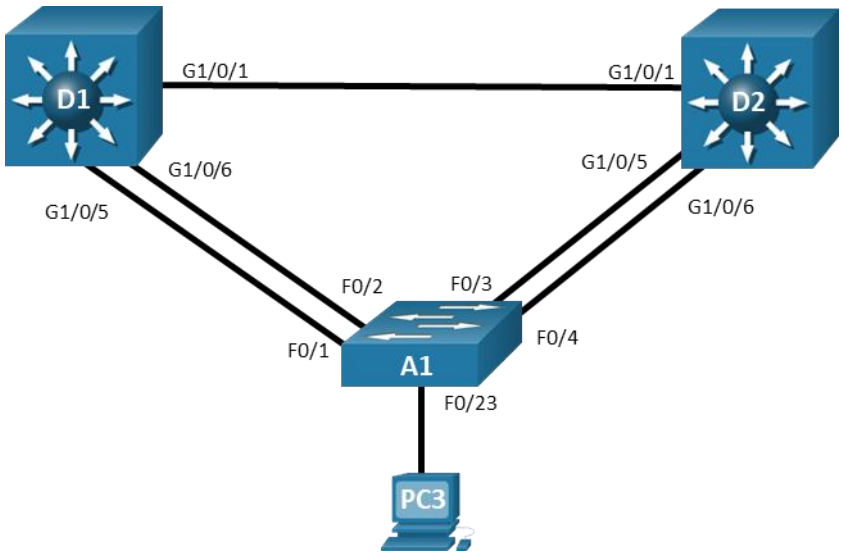
Lab - Implement Advanced STP Modifications and Mechanisms

Topology



Addressing Table

Device	Interface	IP Address
D1	VLAN 1	10.0.0.1/8
D2	VLAN 1	10.0.0.2/8
A1	VLAN 1	10.0.0.3/8

Objectives

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

Part 2: Implement and Observe Various Topology Tuning Methods

Part 3: Implement and Observe Various Topology Protection Mechanisms

Background / Scenario

Although spanning tree works “out of the box”, the default values used in the decisions it makes may lead to logical topologies that, although loop-free, do not align to what you need for your network. In addition, spanning tree “out of the box” is vulnerable to several different scenarios where the root bridge status could be taken over, or a loop could be introduced in the network. In this lab you will configure and observe various ways of bending the logical spanning tree topology to meet your requirements, as well as the different topology protection mechanism that are available. The terms "switch" and "bridge" will be used interchangeably throughout the lab.

Note: This lab is an exercise in deploying and verifying various STP mechanisms and does not reflect networking best practices.

Note: The switches used with CCNP hands-on labs are Cisco 3650 with Cisco IOS XE release 16.9.4 (universalk9 image) and Cisco 2960+ with IOS release 15.2 (lanbase image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Required Resources

- 2 Switches (Cisco 3650 with Cisco IOS XE release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 2960+ with Cisco IOS release 15.2 lanbase image or comparable)
- 1 PC (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings and Interface Addressing

In Part 1, you will set up the network topology and configure basic settings and interface addressing on routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each switch.

a. Console into each switch, enter global configuration mode, and apply the basic settings and interface addressing. The startup configuration is provided below for each switch in the topology.

Switch D1

```
hostname D1
banner motd # D1, STP Tuning and Protection #
spanning-tree mode rapid-pvst
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface range g1/0/1, g1/0/5-6
 switchport mode trunk
 no shutdown
 exit
vlan 2
 name SecondVLAN
```

```
 exit
interface vlan 1
 ip address 10.0.0.1 255.0.0.0
 no shut
 exit
```

Switch D2

```
hostname D2
banner motd # D2, STP Tuning and Protection #
spanning-tree mode rapid-pvst
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface range g1/0/1, g1/0/5-6
 switchport mode trunk
 no shutdown
 exit
vlan 2
 name SecondVLAN
 exit
interface vlan 1
 ip address 10.0.0.2 255.0.0.0
 no shut
 exit
```

Switch A1

```
hostname A1
banner motd # A1, STP Tuning and Protection #
spanning-tree mode rapid-pvst
 line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range f0/1-24, g0/1-2
 shutdown
 exit
interface range f0/1-4
 switchport mode trunk
 no shutdown
 exit
vlan 2
 name SecondVLAN
```

```
 exit
interface vlan 1
 ip address 10.0.0.3 255.0.0.0
 no shut
 exit
```

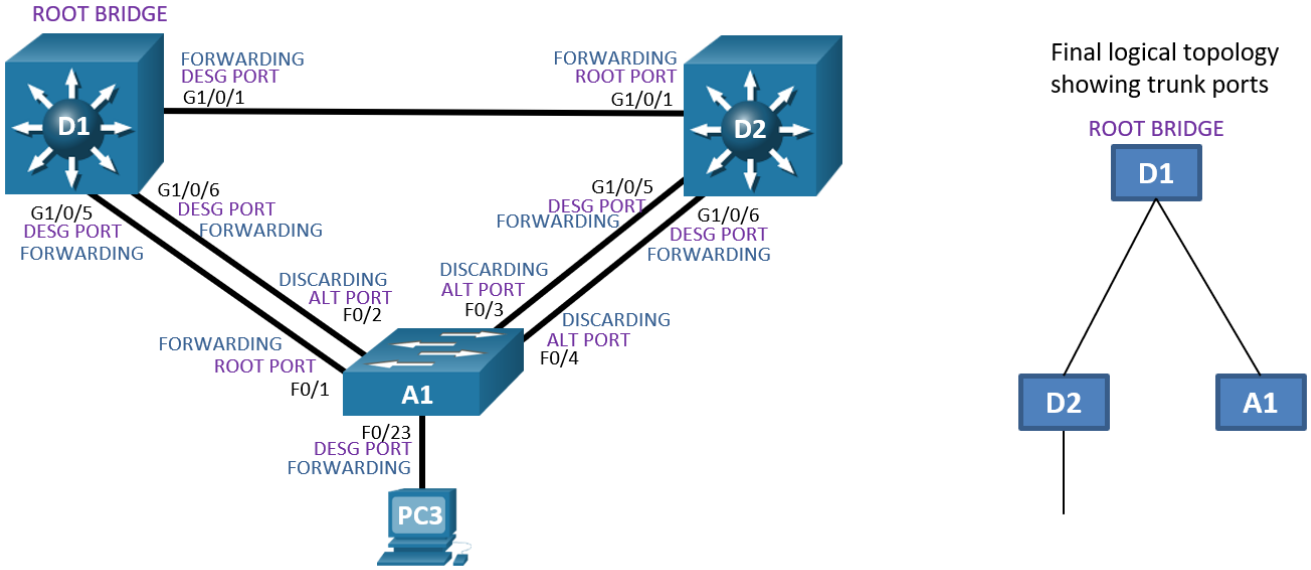
b. Set the clock on each switch to UTC time.

c. Save the running configuration to startup-config.

Note: Outputs and Spanning Tree topologies highlighted in this lab may be different than what you observe using your own equipment. It is critically important for you to understand how Spanning Tree makes its decisions, and how those decisions impact the operational topology of the network.

Step 3: Discover the default spanning tree.

Your switches have been configured, interfaces have been enabled, and Rapid Spanning Tree has already converged onto a loop-free logical network. Before proceeding with the lab, you need to be informed of what the spanning tree topology looks like. You need to know where the root bridge is and where the root, designated, and alternate ports are on each segment for each VLAN. It may be helpful to draw this information out. The image below details the spanning tree operations for the equipment this lab was created on. The spanning tree topology is the same for both VLAN 1 and VLAN 2.



Part 2: Implement and Observe Various Topology Tuning Methods

In Part 2, you will implement various topology tuning methods.

Note: For this part of the lab, PC 3 is turned off and A1 interface F0/23 is not participating in STP.

Step 1: Controlling the Root Bridge.

The current root bridge was elected based on the lowest Bridge ID (consisting of the Priority, extended system ID equal to the VLAN ID, and base MAC address values).

With the priority and extended system IDs being identical, the root bridge's MAC is numerically smaller than the local bridge's MAC. The result is that in a completely un-configured network, one single switch will be elected as the root bridge. The resulting choice of switch may or may not be desirable.

There are two basic ways to manipulate the configuration to control the location of the root bridge:

- The **spanning-tree vlan** *vlan-id* **priority** *value* command can be used to manually set a priority value
- The **spanning-tree vlan** *vlan-id* **root** { **primary** | **secondary** } command can be used to automatically set a priority value.

The difference between the two is that the **priority** command will set a specific number (multiple of 4096) as the priority. This number must be an increment of 4096. The **root primary** command will set the local bridge's priority to 24,576 (if the local bridge MAC is lower than the current root bridge's MAC) or 4096 lower than the current root's priority (if the local bridge MAC is higher than the current root bridge's MAC). Notice that 24,576 is the sixth increment of 4096.

The logic behind this operation is straight-forward. The **root primary** command tries to lower the priority only as much as is needed to win the root election, while leaving priorities between 24576 and the default 32768 for use by secondary bridges. The command always takes the entire Bridge ID into account when computing the resulting priority value.

The **spanning-tree vlan** *vlan-id* **root secondary** command will statically set the local bridge's priority to 28,672. In an otherwise unconfigured network where all switch priorities default to 32,768, the **root primary** command will set the priority on the switch to 24,576 (two 4096 increments lower than the default priority) while the **root secondary** command will set the priority on the secondary root to the 28,672 (one 4096 increment lower than the default priority).

a. Modify D1 and D2 so that D1 is elected the primary root bridge for VLAN 1 and D2 is elected the primary root bridge for VLAN 2. D1 should be elected as the secondary root bridge for VLAN 2, and D2 should be elected as the secondary root bridge for VLAN 1. You will need to make configuration changes on both D1 and D2. The commands used at D1 are as follows:

```
D1(config)# spanning-tree vlan 1 root primary
D1(config)# spanning-tree vlan 2 root secondary
```

b. After you have configured both D1 and D2, go to A1 and issue the command **show spanning-tree root**. In this output you will see the root bridges differentiated.

c.

```
A1# show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         28673 d8b1.9028.af80        19    2   20  15  Fa0/1
VLAN0002         24578 d8b1.905d.c300        19    2   20  15  Fa0/3
```

From the above output, you can see that the root port for VLAN 1 is F0/1 and the root port for VLAN 2 is F0/3.

Step 2: Adjust port cost values to impact root and designated port selection.

As the network is implemented right now, there are two direct paths between switch A1 and the root bridge for each VLAN. Path and port costs are evaluated to determine the shortest path to the root bridge. In the case where there are multiple equal cost paths to the root bridge, additional attributes must be evaluated. In our case, the lower interface number (for example, F0/1) is chosen as the Root Port, and the higher interface number (for example, F0/2) is put into a spanning tree Discarding state.

You can see which ports are blocked with the **show spanning-tree** *vlan-id* command or the **show spanning-tree blockedports** command. For now, examine VLAN 1 on D1.

a. On A1, issue the commands **show spanning-tree vlan 1** and **show spanning-tree blockedports**.

```
A1# show spanning-tree vlan 1
```

```
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    28673
              Address     d8b1.9028.af80
              Cost        19
              Port        1 (FastEthernet0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     f078.1647.4580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Root FWD 19        128.1    P2p
Fa0/2               Altn BLK 19        128.2    P2p
Fa0/3               Altn BLK 19        128.3    P2p
Fa0/4               Altn BLK 19        128.4    P2p

A1# show spanning-tree blockedports

Name                 Blocked Interfaces List
-------------------- ------------------------------------
VLAN0001             Fa0/2, Fa0/3, Fa0/4
VLAN0002             Fa0/1, Fa0/2, Fa0/4

Number of blocked ports (segments) in the system : 6
```

As you can see, VLAN 1 has its Root Port on F0/1. F0/2, F0/3, and F0/4 are Alternate Blocking Ports.

To manipulate which port becomes the Root Port on non-root bridges, change the port cost or port priority value. Remember that this change could have an impact on downstream switches as well.

Note: The changes you are about to implement are considered topology changes and *could* have a significant impact on the overall structure of the spanning tree in your switch network. Do not make these changes in a production network without careful planning and prior coordination.

b. On A1, shutdown interfaces F0/1 and F0/2, assign a new port cost to F0/2, and then issue **no shutdown** to the ports.

```
A1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface range f0/1-2
A1(config-if-range)# shutdown
A1(config-if-range)# exit
A1(config)# interface f0/2
A1(config-if)# spanning-tree cost 12
A1(config-if)# exit
A1(config)# interface range f0/1-2
A1(config-if-range)# no shutdown
```

```
A1(config-if-range)# exit
A1(config)# end
```

c. Now verify that this impacts root port selection on A1 using the **show spanning-tree vlan 1** and **show spanning-tree blockedports** commands.

```
A1# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    28673
              Address     d8b1.9028.af80
              Cost        12
              Port        2 (FastEthernet0/2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     f078.1647.4580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
Fa0/1               Altn BLK 19        128.1    P2p
Fa0/2               Root FWD 12        128.2    P2p
Fa0/3               Altn BLK 19        128.3    P2p
Fa0/4               Altn BLK 19        128.4    P2p

A1# show spanning-tree blockedports

Name                 Blocked Interfaces List
-------------------- ------------------------------------
VLAN0001             Fa0/1, Fa0/3, Fa0/4
VLAN0002             Fa0/1, Fa0/3, Fa0/4

Number of blocked ports (segments) in the system : 6
```

From the output you can see that the root port selected by A1 for VLAN 1 is now interface F0/2, and the port (and root) cost is now 12. There is another impact to the cost being set as it has been. Issue the command **show spanning-tree root** on A1.

```
A1# show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         28673 d8b1.9028.af80        12    2   20  15  Fa0/2
VLAN0002         24578 d8b1.905d.c300        16    2   20  15  Fa0/2
```

Notice that the root port for VLAN 2 is now F0/2, instead of F0/3. Why? Because the total path cost to D2 via F0/2 is now 16, which is less than the cost of the direct link to D2 via F0/3 or F0/4.

d. Adjust the cost value of interface F0/2 on A1 to 18. This will make the VLAN 2 root port F0/3 again.

```
A1(config)# interface range f0/1-2
A1(config-if-range)# shutdown
A1(config-if-range)# exit
A1(config)# interface f0/2
A1(config-if)# spanning-tree cost 18
A1(config-if)# exit
A1(config)# interface range f0/1-2
A1(config-if-range)# no shutdown
A1(config-if-range)# exit
A1(config)# end
A1#
A1# show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         28673 d8b1.9028.af80        18    2   20  15  Fa0/2
VLAN0002         24578 d8b1.905d.c300        19    2   20  15  Fa0/3
```

Step 3: Adjust port priority values to impact root port selection.

The next method to impact root port selection is configured on the root bridge itself. In our current network topology, A1 has two connections to the root bridge for VLAN 2, switch D2. The root port has been selected, in this case based on the lowest port ID. Port ID is made up of two values, labeled as Prio (Priority) and Nbr (Number).

Note: The port number is not necessarily equal to the interface ID. A switch may use any port number for STP purposes if they are unique for each port on the switch.

The port priority can be any value between 0 and 240, in increments of 16 (older switches may allow setting the priority in different increments).

a. On A1, issue the command **show spanning-tree vlan 2** and take note of the port ID values listed.

```
A1# show spanning-tree vlan 2

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID     Priority    24578
              Address     d8b1.905d.c300
              Cost        19
              Port        3 (FastEthernet0/3)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32770  (priority 32768 sys-id-ext 2)
              Address     f078.1647.4580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
```

```
Fa0/1               Altn BLK 19        128.1    P2p
Fa0/2               Altn BLK 18        128.2    P2p
Fa0/3               Root FWD 19        128.3    P2p
Fa0/4               Altn BLK 19        128.4    P2p
```

As expected with two equal-cost paths to the root bridge, the lower port ID was selected as the root port.

b. Modify the port priority of D2 interface G1/0/6 so that it becomes the preferred port.

```
D2(config)# interface range g1/0/5-6
D2(config-if-range)# shutdown
D2(config-if-range)# exit
D2(config)# interface g1/0/6
D2(config-if)# spanning-tree port-priority 64
D2(config-if)# exit
D2(config)# interface range g1/0/5-6
D2(config-if-range)# no shutdown
D2(config-if-range)# exit
D2(config)# end
```

c. On A1, issue the **show spanning-tree vlan 2** command and you will see that F0/4 is now the selected root port. This selection is based on the lower priority value of D2 interface G1/0/6. Notice that the lower priority value does not appear in any A1 output.

```
A1# show spanning-tree vlan 2

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID     Priority    24578
              Address     d8b1.905d.c300
              Cost        19
              Port        4 (FastEthernet0/4)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32770  (priority 32768 sys-id-ext 2)
              Address     f078.1647.4580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/1               Altn BLK 19        128.1    P2p
Fa0/2               Altn BLK 18        128.2    P2p
Fa0/3               Altn BLK 19        128.3    P2p
Fa0/4               Root FWD 19        128.4    P2p
```

Step 4: Implement Spanning Tree Portfast.

In both STP and RSTP, a newly connected port must be guaranteed not to create a switching loop before it can become a Forwarding port. This may take up to 30 seconds. However, such a check is not necessary for ports connected to end devices that do not perform switching or bridging, such as workstations, network printers, servers, etc. In RSTP, these ports are called *edge* ports (ports that connect to other switches in the

topology are called *non-edge* ports). Edge ports can safely enter the Forwarding state right after they come up, because they do not connect to any device capable of forwarding frames.

Cisco developed a feature called PortFast that essentially allow you to define a port as an edge port. Any PortFast-enabled port will enter the Forwarding state immediately after coming up, without going through the intermediary non-forwarding states, saving 30 seconds each time a new connection is made to the port. PortFast can be used with all STP versions.

Apart from allowing a port to jump into the Forwarding state as soon as it is connected, the concept of an edge port is extremely important in RSTP and MSTP. Recall that as part of its improvements over legacy STP, RSTP uses a Proposal/Agreement mechanism to rapidly, yet safely enable a link between switches if one of the switches has its Root port on that link.

Upon receiving a Proposal on its Root port, a switch puts all its non-edge Designated ports into the Discarding state, effectively cutting itself off from the network and preventing a possible switching loop (this is called the Sync operation). When this is accomplished, the switch sends an Agreement back out its Root port so that the upstream Designated port receiving this Agreement can be immediately put into the Forwarding state. The switch will then start sending its own Proposals on all its non-edge Designated ports that have just become Discarding. It will wait for Agreements to arrive from downstream switches which will allow these ports to instantaneously become Forwarding again.

If end devices are connected to ports not configured as edge (that is, PortFast) ports, these ports will become Discarding during the Sync operation. Because end hosts do not support RSTP and cannot send back an Agreement, they will be cut off from the network for 30 seconds until the ports reach the Forwarding state using ordinary timers. As a result, users will experience significant connectivity outages.

Ports configured as edge ports are not affected by the Sync operation and will remain in the Forwarding state even during the Proposal/Agreement handling. Activating RSTP in a network without properly configuring ports toward end hosts as edge ports will cause the network to perform possibly even more poorly than with legacy STP. With RSTP, proper configuration of ports toward end hosts as edge ports is critical. Cisco switches default to all their ports being non-edge ports.

a. Ensure that PC 3 is turned on.

b. On A1, issue the command **debug spanning-tree events**, then issue the **no shutdown** command for interface F0/23, wait a few seconds, and issue the **shutdown** command, followed by **end** and **undebug all**. The log output will appear something like the output below.

```
A1# debug spanning-tree events
Spanning Tree event debugging is on
A1#
A1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface f0/23
A1(config-if)# no shutdown
A1(config-if)#
Dec 24 17:32:55.461: RSTP(1): initializing port Fa0/23
Dec 24 17:32:55.461: RSTP(1): Fa0/23 is now designated
Dec 24 17:32:55.469: RSTP(1): transmitting a proposal on Fa0/23
A1(config-if)#
Dec 24 17:32:55.813: %LINK-3-UPDOWN: Interface FastEthernet0/23, changed state to up
Dec 24 17:32:55.838: RSTP(1): transmitting a proposal on Fa0/23
A1(config-if)#
Dec 24 17:32:56.820: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to up
A1(config-if)#
Dec 24 17:32:57.852: RSTP(1): transmitting a proposal on Fa0/23
```

```
A1(config-if)# shutdown
A1(config-if)#
Dec 24 17:32:59.873: RSTP(1): transmitting a proposal on Fa0/23
A1(config-if)#
Dec 24 17:33:03.807: %LINK-5-CHANGED: Interface FastEthernet0/23, changed state to
administratively down
Dec 24 17:33:04.814: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to down
A1(config-if)# end
A1#un a
Dec 24 17:33:08.530: %SYS-5-CONFIG_I: Configured from console by console
A1# undebug all
All possible debugging has been turned off
```

What you see here is the switch trying to go through the Proposal/Agreement process on F0/23. But there is no point in this because the device connected to F0/23 is an endpoint and does not understand Spanning Tree. This adds the potential of a 30-second delay before the host can send data, such as a DHCP request to the network.

c. On A1, issue the command **debug spanning-tree events**, then configure interface F0/23 with the **spanning-tree portfast** command followed by the **no shutdown** command. This designates F0/23 as an interface that will never be connected to another switch, and therefore; it will never cause a loop in the topology, and subsequently allow that interface to go into forwarding mode immediately. Observe the output.

```
A1# debug spanning-tree events
Spanning Tree event debugging is on
A1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface f0/23
A1(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on FastEthernet0/23 but will only
 have effect when the interface is in a non-trunking mode.
A1(config-if)# no shutdown
A1(config-if)# exit
A1(config)#
A1(config)#
A1(config)#
Dec 24 17:39:40.941: RSTP(1): initializing port Fa0/23
Dec 24 17:39:40.941: RSTP(1): Fa0/23 is now designated
A1(config)#
Dec 24 17:39:41.318: %LINK-3-UPDOWN: Interface FastEthernet0/23, changed state to up
Dec 24 17:39:42.325: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to up
```

From the output you can see that RSTP sees F0/23 as designated, and never sends a proposal on the interface, because of the portfast setting.

There are two other ways to configure an interface as a portfast port; using the **switchport host** interface configuration command and using the **spanning-tree portfast default** global configuration command.

- **switchport host** not only enables portfast, but also statically sets the interface mode to access and disables aggregation protocols.
- The **spanning-tree portfast default** command sets the default state of interfaces that are configured as access ports with portfast enabled. All you must do is configure the interface with **switchport mode access** and portfast is engaged on that interface.

Verifying that a port is in portfast mode can be done by looking at the running-configuration for that port or by examining spanning-tree details for the port. For example, use the **show spanning-tree interface** *interface-id* command to verify that the interface is in Edge mode, as shown below:

```
A1# show spanning-tree interface f0/23

Vlan                Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
VLAN0001            Desg FWD 100       128.23   P2p Edge
```

Or, issue the command **show spanning-tree detail** | **section FastEthernet0/23**, as shown below:

```
A1# show spanning-tree detail | section FastEthernet0/23
          from FastEthernet0/23
 Port 23 (FastEthernet0/23) of VLAN0001 is designated forwarding
   Port path cost 100, Port priority 128, Port Identifier 128.23.
   Designated root has priority 28673, address d8b1.9028.af80
   Designated bridge has priority 32769, address f078.1647.4580
   Designated port id is 128.23, designated path cost 18
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast mode
   Link type is point-to-point by default
   BPDU: sent 371212, received 0
```

Part 3: Implement and Observe Various Topology Protection Mechanisms

In this part of the lab, you will implement and observe topology protection mechanisms such as root guard, bpdu guard, bpdu filter, and loop guard.

Step 1: Implement and observe Root Guard.

Root Guard helps prevent a root switch or Root Port takeover. It is configured on the port that is to be protected. If a port that is protected by Root Guard receives a superior BPDU that would normally cause the port to become a Root Port, the BPDU will be discarded and the port will be moved to the Root-Inconsistent state. An STP inconsistent state differs from the error disabled state in that the port is not disabled entirely; instead, it is only put into the Blocking (Discarding) state and will be put back into its proper role and state once the cause for its inconsistent state disappears. With Root Guard, a port will be reinstated into its appropriate role and state automatically when it stops receiving superior BPDUs.

Note: BPDU Root Guard is a protective mechanism used in situations where your network and the network of your customer need to form a single STP domain, yet you want to have the STP root bridge in your portion of the network and you do not want your customer to take over this root switch selection, or back up the connectivity in your network through the customer. In these cases, you would put the Root Guard on ports toward the customer. However, inside your own network, using Root Guard would be harmful. Your network can be considered trustworthy and there is no rogue root switch to protect against. Using Root Guard in your own network would cause it to be unable to converge on a new workable spanning tree if any of the primary

links failed, and it would also prevent your network from converging to a secondary root switch if the primary root switch failed entirely.

a. To illustrate the behavior of Root Guard, we will configure it on a designated port on D2 for VLAN 2. D2 is the root bridge for VLAN 2, so all trunk ports are designated.

```
D2# show span root

                                        Root    Hello Max Fwd
Vlan                 Root ID            Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001         28673 d8b1.9028.af80         4    2   20  15  Gi1/0/1
VLAN0002         24578 d8b1.905d.c300         0    2   20  15

D2# show spanning-tree detail | include VLAN0002
 VLAN0002 is executing the rstp compatible Spanning Tree protocol
 Port 1 (GigabitEthernet1/0/1) of VLAN0002 is designated forwarding
 Port 5 (GigabitEthernet1/0/5) of VLAN0002 is designated forwarding
 Port 6 (GigabitEthernet1/0/6) of VLAN0002 is designated forwarding
```

b. Go to A1 and verify what is the root port for VLAN0002. It should be interface F0/4 because of the change in port priority we configured earlier on D2.

```
A1# show span root | include VLAN0002
VLAN0002         24578 d8b1.905d.c300        19    2   20  15  Fa0/4
```

c. Now on D2, add root guard to the ports connected to A1.

```
D2(config)# interface range g1/0/5-6
D2(config-if-range)# spanning-tree guard root
*Jan  2 14:02:07.785: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port GigabitEthernet1/0/5.
*Jan  2 14:02:07.792: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port GigabitEthernet1/0/6.
D2(config-if-range)# exit
```

d. To verify that root guard is working, try to have A1 take over as root bridge for VLAN0002. Issue the command **spanning-tree vlan 2 priority 16384**.

```
A1(config)# spanning-tree vlan 2 priority 16384
```

e. Return to D1 and issue the command show spanning-tree vlan 2.

```
D2# show spanning-tree vlan 2

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID     Priority    16386
              Address     f078.1647.4580
              Cost        23
              Port        1 (GigabitEthernet1/0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24578  (priority 24576 sys-id-ext 2)
              Address     d8b1.905d.c300
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/5             Desg BKN*19        128.5    P2p *ROOT_Inc
Gi1/0/6             Desg BKN*19         64.6    P2p *ROOT_Inc
```

This output has two indicators of the issue. First BKN* is short for "BROKEN", and *ROOT_Inc represents the Root Inconsistent message. A list of all STP inconsistent ports including the reason for their inconsistency can also be requested with the command **show spanning-tree inconsistentports**.

```
D2# show spanning-tree inconsistentports

Name                 Interface                     Inconsistency
-------------------- ----------------------------- ------------------
VLAN0002             GigabitEthernet1/0/5          Root Inconsistent
VLAN0002             GigabitEthernet1/0/6          Root Inconsistent

Number of inconsistent ports (segments) in the system : 2
```

f. To return things to normal, issue the command **no spanning-tree vlan 2 priority 16384** on A1 and then remove root guard on interfaces G1/0/5 and G1/0/6 of D2 with the command **no spanning-tree guard root**.

Step 2: Implement and observe BPDU Guard.

PortFast causes an interface to go into Forwarding state immediately. There is a risk that if two PortFast-enabled ports are inadvertently or maliciously connected, they will both come up as Forwarding ports, immediately creating a switching loop.

The default expected behavior of a PortFast port that receives a BPDU is for that port to revert to a normal spanning-tree non-edge port. There is the potential that the load on a given switch might be too great to handle the received BPDU properly, prolonging the loop condition.

BPDU Guard adds another layer of protection. Whenever a port protected by BPDU Guard unexpectedly receives a BPDU, it is immediately put into err-disabled state. Any interface can be protected with BPDU Guard, but its typical use is on PortFast-enabled ports.

BPDU Guard can be configured globally or on a per-interface basis. If the BPDU Guard is configured on the global level using the **spanning-tree portfast bpduguard default** command, the BPDU Guard will be automatically enabled on all PortFast-enabled ports of the switch. If the BPDU Guard is configured on an interface using the **spanning-tree bpduguard enable** command, it will apply to this port unconditionally, regardless of whether it is a PortFast-enabled port.

For this example, we will configure BPDU guard on a trunking interface that is a non-root port on A1. Configuring BPDU Guard on an interface that is intended to be a trunk is *not a recommended practice*; we are doing this just to demonstrate the functionality of the tool.

a. Verify the trunking ports and root ports on A1 using the commands **show spanning-tree root** and **show interface trunk**. From the output below, we see that interface F0/1 will meet the requirements for this demonstration (non-root trunk).

```
A1# show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/1       on               802.1q         trunking      1
Fa0/2       on               802.1q         trunking      1
Fa0/3       on               802.1q         trunking      1
```

```
Fa0/4       on              802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1-4094
Fa0/2       1-4094
Fa0/3       1-4094
Fa0/4       1-4094

Port        Vlans allowed and active in management domain
Fa0/1       1-2
Fa0/2       1-2
Fa0/3       1-2
Fa0/4       1-2

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       none
Fa0/2       1
Fa0/3       none
Fa0/4       2

A1# show span root

                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  -----------
VLAN0001         28673 d8b1.9028.af80        18    2   20  15  Fa0/2
VLAN0002         24578 d8b1.905d.c300        19    2   20  15  Fa0/4
```

b. On A1 interface F0/1, issue the command **spanning-tree bpduguard enable**. As you can see, the interface is almost immediately err-disabled. Issue the **shutdown** command, remove BPDU Guard with the **no spanning-tree bpduguard enable** command, and issue the **no shutdown** command on interface F0/1 to bring it back up. Verify the trunk is operational with the **show interface trunk** command.

```
A1(config)# interface f0/1
A1(config-if)# spanning-tree bpduguard enable
A1(config-if)#
Jan  2 15:19:11.899: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port Fa0/1 with
BPDU Guard enabled. Disabling port.
A1(config-if)#
Jan  2 15:19:11.899: %PM-4-ERR_DISABLE: bpduguard error detected on Fa0/1, putting
Fa0/1 in err-disable state
Jan  2 15:19:12.905: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to down
A1(config-if)#
Jan  2 15:19:13.920: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
A1(config-if)# shutdown
A1(config-if)#
Jan  2 15:19:22.955: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down
A1(config-if)# no spanning-tree bpduguard enable
A1(config-if)# no shutdown
```

```
Jan  2 15:19:39.950: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
A1(config-if)#
Jan  2 15:19:43.566: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to up
A1(config-if)# end

A1# show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/1       on               802.1q         trunking      1
Fa0/2       on               802.1q         trunking      1
Fa0/3       on               802.1q         trunking      1
Fa0/4       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1-4094
Fa0/2       1-4094
Fa0/3       1-4094
Fa0/4       1-4094

Port        Vlans allowed and active in management domain
Fa0/1       1-2
Fa0/2       1-2
Fa0/3       1-2
Fa0/4       1-2

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       none
Fa0/2       1
Fa0/3       none
Fa0/4       2
```

Step 3: Implement and observe BPDU Filter.

Neither PortFast nor BPDU Guard prevents the switch from sending BPDUs on an interface; if such a behavior is required, BPDU Filter can be used. It can be configured either globally or at a specific interface.

If BPDU Filter is configured on the global level using the **spanning-tree portfast bpdufilter default** global configuration command, the BPDU Filter applies only to PortFast-enabled ports. When these ports come up, they will *send* up to 11 BPDUs and then stop sending further BPDUs. If the BPDU Filter-configured interface *receives* a BPDU *at any time*, the BPDU Filter and PortFast will be deactivated on that port and it will become a normal spanning tree interface. As a result, a globally configured BPDU Filter does not prevent ports from receiving and processing BPDUs; it only attempts to stop sending BPDUs on ports where most probably, there is no device attached that would process them.

If you configure an interface with the **spanning-tree bpdufilter enable** command, the port will stop sending and processing received BPDUs altogether. This can be used, for example, to split a network into two or more independent STP domains, each having its own root bridge and resulting topology. However, because these domains are no longer protected against mutual loops by STP, it is the task of the network administrator to make sure that these two domains are never connected by more than just a single link.

For this demonstration, we will configure BPDU filter at the interface level.

a. Find out how may BPDUs interface F0/23 on Switch A1 has sent using the **show spanning-tree interface f0/23 detail | i BPDU** command. Repeat the command several times to validate that the BPDU count is increasing.

```
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374695, received 0
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374696, received 0
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374697, received 0
```

b. Configure the interface with BPDU filter using the **spanning-tree bpdufilter enable** command.

```
A1# config t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface f0/23
A1(config-if)# spanning-tree bpdufilter enable
A1(config-if)# exit
A1(config)# end
```

c. Verify BPDUs are no longer being sent. Issue the command **show spanning-tree interface f0/23 detail | i BPDU** several times and you should see that the BPDU count is not increasing.

```
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374726, received 0
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374726, received 0
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374726, received 0
```

d. Remove BPDU filter with the **no spanning-tree bpdufilter enable** command.

```
A1# config t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface f0/23
A1(config-if)# no spanning-tree bpdufilter enable
A1(config-if)# exit
A1(config)# end
```

e. Verify that BPDUs are now being sent.

```
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374745, received 0
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374746, received 0
A1# show spanning-tree interface f0/23 detail | i BPDU
   BPDU: sent 374747, received 0
```

Step 4: Implement and observe Loop Guard.

Loop Guard prevents Root and Alternate ports from becoming Designated ports if BPDUs suddenly stop being received on them.

In a normal STP network, all ports receive and process BPDUs, even Blocking (Discarding) ports. This is how they know that the device at the other end of the link is alive and still superior to them. If a Blocked port stops

receiving these BPDUs, it can only assume that the device on the other side is no longer present and they are now superior and should be in Forwarding state for the given segment. An example of when this could occur is the instance where the Rx fiber in an optical cable becomes disconnected, cut, or connected to a different port or device than the corresponding Tx fiber, creating a uni-directional link.

This could cause permanent switching loops in the network, so Loop Guard helps to prevent them.

Loop Guard can be enabled globally using the **spanning-tree loopguard default global** configuration command, or on a per-interface basis using the **spanning-tree guard loop** command. Loop Guard should never be enabled on PortFast-enabled ports

For this example, we will configure Loop Guard on an Alternate port on A1, and then stop sending out BPDUs from the corresponding Designated port on the other end of the link.

a. Verify which ports are Alternate ports for VLAN 2 on A1 using the **show spanning-tree vlan 2** command.

```
A1# show spanning-tree vlan 2

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID     Priority    24578
              Address     d8b1.905d.c300
              Cost        19
              Port        4 (FastEthernet0/4)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32770  (priority 32768 sys-id-ext 2)
              Address     f078.1647.4580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
Fa0/1               Altn BLK 19        128.1    P2p
Fa0/2               Altn BLK 18        128.2    P2p
Fa0/3               Altn BLK 19        128.3    P2p
Fa0/4               Root FWD 19        128.4    P2p
```

b. On A1, configure interface F0/1 with loop guard.

```
A1# config t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface f0/1
A1(config-if)# spanning-tree guard loop
A1(config-if)# exit
A1(config)# end
```

c. On D1, configure the port connecting to F0/1 for bpdufilter; in this topology it is interface G1/0/5.

```
D1(config)# interface g1/0/5
D1(config-if)# spanning-tree bpdufilter enable
```

d. On A1, you should receive a SYSLOG message stating that Loop Guard has blocked port F0/1. Issue the command **show spanning-tree vlan 2** and you will see that F0/1 is broken. Issue the command **show spanning-tree inconsistentports** and you will see that F0/1 is loop-inconsistent.

```
A1#
```

```
Jan  2 16:23:56.915: %SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port
FastEthernet0/1 on VLAN0002.
A1#
A1# show spanning-tree vlan 2

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID     Priority    24578
              Address     d8b1.905d.c300
              Cost        19
              Port        4 (FastEthernet0/4)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32770  (priority 32768 sys-id-ext 2)
              Address     f078.1647.4580
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
Fa0/1               Desg BKN*19        128.1    P2p *LOOP_Inc
Fa0/2               Altn BLK 18        128.2    P2p
Fa0/3               Altn BLK 19        128.3    P2p
Fa0/4               Root FWD 19        128.4    P2p

A1# show spanning-tree inconsistentports

Name                 Interface               Inconsistency
-------------------- ----------------------- ------------------
VLAN0001             FastEthernet0/1         Loop Inconsistent
VLAN0002             FastEthernet0/1         Loop Inconsistent

Number of inconsistent ports (segments) in the system : 2
```

e. On D1, remove the BPDU filter on interface G1/0/5.

```
D1(config)# interface g1/0/5
D1(config-if)# no spanning-tree bpdufilter enable
D1(config-if)#
```

f. On A1, you should see a SYSLOG message indicating Loop Guard has removed the block on interface F0/1. Remove the loop guard configuration.

```
A1#
Jan  2 16:28:05.075: %SPANTREE-2-LOOPGUARD_UNBLOCK: Loop guard unblocking
port FastEthernet0/1 on VLAN0001.
A1# show spanning-tree inconsistentports

Name                 Interface               Inconsistency
-------------------- ----------------------- ------------------
```

```
Number of inconsistent ports (segments) in the system : 0

A1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)# interface f0/1
A1(config-if)# no spanning-tree guard loop
A1(config-if)# end
```