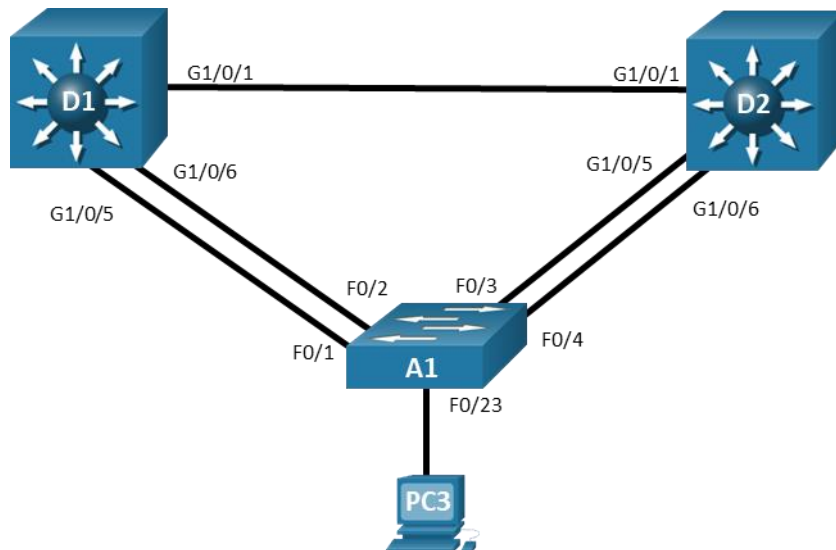


## Lab - Implement MST (Instructor Version)

**Instructor Note:** Red font color or gray highlights indicate text that appears in the instructor copy only.

### Answers: [4.1.2 Lab - Implement MST](#)

#### Topology



#### Objectives

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

**Part 2: Implement and Observe MST**

**Part 3: Configure, Tune and Verify Basic MST Operation**

#### Background / Scenario

Cisco's Per VLAN Spanning Tree (PVST+) provides a significant step up from standard spanning tree in terms of flexibility. PVST+ allows each VLAN to have its own independent spanning tree, thereby making better use of available links in the network. A drawback to PVST+ is that there is an instance of PVST+ running for each VLAN in the network, regardless of whether there are different spanning-tree topologies required. This presents the potential for overwhelming the switch CPU and memory. Additionally, Cisco switches support a finite number of PVST instances. If more VLANs are created than there are PVST+ instances supported on a particular switch, some of the VLANs will not have any STP running, and therefore not have any switching loop protection. PVST+ and Rapid PVST+ are simply unusable in that kind of environment. Lastly, PVST+ and Rapid PVST+ are Cisco-proprietary protocols and generally unusable in mixed vendor environments.

Cisco was involved in the early development of Multiple Spanning Tree. MST was standardized as IEEE 802.1s in 2002 and merged into 802.1Q in 2005. MST is an open protocol derived from RSTP, sharing all its rapid convergence properties, and in fact, the only standardized spanning-tree protocol for VLAN-based networks supported by multiple vendors. MST is a compromise between common spanning-tree and per-VLAN spanning tree. An MST instance represents a unique spanning-tree topology. Multiple MST instances

can be created to account for each of the required spanning-tree topologies in a network, and an arbitrary number of VLANs can be mapped to a single MST instance.

In this lab, you will set up two instances of MST, one for VLANs 2 and 3 and the other for VLANs 4 and 5. All other VLANs will be mapped to the default MST instance (also referred to as IST or Internal Spanning Tree).

**Note:** This lab is an exercise in deploying and verifying MST 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.

**Instructor Note:** Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

### 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.

##### 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. The startup configurations for each device are provided below.

###### Switch D1

```
hostname D1
banner motd # D1, Multiple Spanning Tree #
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
vlan 3
    name ThirdVLAN
    exit
vlan 4
    name FourthVLAN
    exit
vlan 5
    name FifthVLAN
    exit
```

### Switch D2

```
hostname D2
banner motd # D2, Multiple Spanning Tree #
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
vlan 3
    name ThirdVLAN
    exit
vlan 4
    name FourthVLAN
    exit
vlan 5
    name FifthVLAN
    exit
```

### Switch A1

```
hostname A1
```

```
banner motd # A1, Multiple Spanning Tree #
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
interface f0/23
  no shutdown
  exit
vlan 2
  name SecondVLAN
  exit
vlan 3
  name ThirdVLAN
  exit
vlan 4
  name FourthVLAN
  exit
vlan 5
  name FifthVLAN
  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.

## Part 2: Implement and Observe MST

### Step 1: Configure MST on D1 and D2.

In this step we will configure MST on D1 and D2 only. We will do this so we can observe their interaction with each other as well as their interaction with A1.

- a. On D1 and D2, issue the command **spanning-tree mode mst**.

```
D1(config)# spanning-tree mode mst
```

- b. At this point, with no MST-specific configuration, MST Instance 0 is operational for all VLANs. Issue the command **show spanning-tree** and you will see in the output that the spanning tree information is about MST 0. Issue the command **show spanning-tree mst** and you will see the MST-specific STP information that is specific to MST 0 only. Take note of the information displayed for interfaces g1/0/5 and g1/0/6

## Lab - Implement MST

because they are connected to a switch that is not running MST. Their type is listed as P2p Bound (PVST).

```
D1# show spanning-tree
```

```
MST0
```

```
Spanning tree enabled protocol mstp
Root ID    Priority    32768
           Address    d8b1.9028.af80
           This bridge is the root
           Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
Bridge ID  Priority    32768 (priority 32768 sys-id-ext 0)
           Address    d8b1.9028.af80
           Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/1	Desg	FWD	20000	128.1	P2p
Gi1/0/5	Desg	FWD	200000	128.5	P2p Bound (PVST)
Gi1/0/6	Desg	FWD	200000	128.6	P2p Bound (PVST)

```
D1# show spanning-tree mst
```

```
##### MST0    vlans mapped:    1-4094
Bridge          address d8b1.9028.af80  priority          32768 (32768 sysid 0)
Root            this switch for the CIST
Operational     hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured      hello time 2 , forward delay 15, max age 20, max hops    20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/1	Desg	FWD	20000	128.1	P2p
Gi1/0/5	Desg	FWD	200000	128.5	P2p Bound (PVST)
Gi1/0/6	Desg	FWD	200000	128.6	P2p Bound (PVST)

- c. The basic behavior of MST is the same as spanning tree, where a root bridge must be selected, then root ports, and finally best paths to the root bridge from all non-root bridges. In the current network, we can see that D1 has been elected the root bridge. The bridge priority defaults to 32768, so the election is based on D1 having a lower base MAC address. The switch elected as root may be different in your lab topology, but the rules for election remain the same. Issue the command **show spanning-tree root** on switch A1. Switch A1 is running five instances of spanning tree.

```
A1# show spanning-tree root
```

Vlan	Root ID	Root Cost	Hello Time	Max Age	Fwd Dly	Root Port
VLAN0001	32768 d8b1.9028.af80	19	2	20	15	Fa0/1
VLAN0002	32768 d8b1.9028.af80	19	2	20	15	Fa0/1
VLAN0003	32768 d8b1.9028.af80	19	2	20	15	Fa0/1

## Lab - Implement MST

```
VLAN0004      32768 d8b1.9028.af80      19    2    20   15   Fa0/1
VLAN0005      32768 d8b1.9028.af80      19    2    20   15   Fa0/1
```

- d. Issue the **show spanning-tree root** command on D2 and the output will be different. This is because with MST, only one instance of the spanning-tree algorithm runs, regardless of the number of VLANs mapped to it.

```
D2# show spanning-tree root
```

```

                Root      Hello Max Fwd
MST Instance    Root ID    Cost   Time  Age Dly  Root Port
-----
MST0            32768 d8b1.9028.af80    0     2   20  15   Gi1/0/1
```

- e. Configure A1 to use MST.

```
A1(config)# spanning-tree mode mst
```

### Part 3: Configure, Tune and Verify Basic MST Operation

In the last part, you configured all three switches to run MST. In this part, you will further configure, tune, and verify MST to support the unique topological requirements.

#### Step 1: Create and verify an MST configuration.

MST allows network engineers to reduce the load of the spanning-tree protocol while still providing unique spanning-tree topologies for groups of VLANs. MST configuration must be completed by hand on each switch individually.

Complete the following tasks on switch D1:

- a. Enter MST configuration mode using the command **spanning-tree mst configuration**.

```
D1(config)# spanning-tree mst configuration
```

- b. Configure an MST region name. Our example will be CCNPv8.

```
D1(config-mst)# name CCNPv8
```

- c. Configure an MST configuration revision number. Our example will be 1.

```
D1(config-mst)# revision 1
```

- d. Configure instance 1 to include VLAN 2.

```
D1(config-mst)# instance 1 vlan 2
```

- e. Configure instance 2 to include VLAN 4.

```
D1(config-mst)# instance 2 vlan 4
```

- f. Commit the configuration by typing exit and returning to global configuration mode.

```
D1(config-mst)# exit
```

```
D1(config)# end
```

- g. Issue the command **show spanning-tree mst** to verify the configuration is in place.

**Note:** While in spanning-tree mst configuration mode, you can use the **show current** and **show pending** commands to view current and pending configuration settings.

```
D1# show spanning-tree mst
```

```
##### MST0      vlans mapped:    1,3,5-4094
```

## Lab - Implement MST

```
Bridge      address d8b1.9028.af80  priority      32768 (32768 sysid 0)
Root        this switch for the CIST
Operational hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured  hello time 2 , forward delay 15, max age 20, max hops 20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	---	-----	-----	-----
Gi1/0/1	Desg	FWD	20000	128.1	P2p
Gi1/0/5	Desg	FWD	200000	128.5	P2p
Gi1/0/6	Desg	FWD	200000	128.6	P2p

```
##### MST1  vlans mapped: 2
Bridge      address d8b1.9028.af80  priority      32769 (32768 sysid 1)
Root        this switch for MST1
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	---	-----	-----	-----
Gi1/0/1	Desg	FWD	20000	128.1	P2p
Gi1/0/5	Desg	FWD	200000	128.5	P2p
Gi1/0/6	Desg	FWD	200000	128.6	P2p

```
##### MST2  vlans mapped: 4
Bridge      address d8b1.9028.af80  priority      32770 (32768 sysid 2)
Root        this switch for MST2
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	---	-----	-----	-----
Gi1/0/1	Desg	FWD	20000	128.1	P2p
Gi1/0/5	Desg	FWD	200000	128.5	P2p
Gi1/0/6	Desg	FWD	200000	128.6	P2p

- h. This configuration does not propagate to other switches. Each switch exchanges digest information summarizing the VLAN-to-Instance mappings it has configured. If a switch receives a BPDU with a different digest, it assumes that the sender is in a different MST region. The output below is what A1 shows in the topology used to create this lab. Notice that the ports connected to D1 are classified as P2p Bound (RSTP).

```
A1# show spanning-tree mst
```

```
##### MST0  vlans mapped: 1-4094
Bridge      address f078.1647.4580  priority      32768 (32768 sysid 0)
Root        address d8b1.9028.af80  priority      32768 (32768 sysid 0)
            port      Fa0/3          path cost    20000
Regional Root address d8b1.905d.c300  priority      32768 (32768 sysid 0)
            internal cost 200000    rem hops 19
Operational hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured  hello time 2 , forward delay 15, max age 20, max hops 20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	---	-----	-----	-----
Fa0/1	Altn	BLK	200000	128.1	P2p Bound(RSTP)

## Lab - Implement MST

---

```
Fa0/2          Altn BLK 200000    128.2    P2p Bound (RSTP)
Fa0/3          Root FWD 200000    128.3    P2p
Fa0/4          Altn BLK 200000    128.4    P2p
Fa0/23         Desg FWD 2000000   128.23   P2p
```

```
A1# show spanning-tree mst configuration digest
```

```
Name          []
Revision 0     Instances configured 1
Digest         0xAC36177F50283CD4B83821D8AB26DE62
Pre-std Digest 0xBB3B6C15EF8D089BB55ED10D24DF44DE
```

And here is the digest from D1:

```
D1# show spanning-tree mst configuration digest
```

```
Name          [CCNPv8]
Revision 1     Instances configured 3
Digest         0xB41829F9030A054FB74EF7A8587FF58D
Pre-std Digest 0x6C5B0D56DA63408ED96AA5F1C5356F98
```

As you can see here, the digest values are different.

- i. Configure MST on D1, change the revision number to 2 and add VLAN 3 to instance 1 and VLAN 5 to instance 2.

```
D1(config)# spanning-tree mst configuration
```

```
D1(config-mst)# show current
```

```
Current MST configuration
Name          [CCNPv8]
Revision 1     Instances configured 3
```

```
Instance Vlans mapped
```

```
-----
0         1,3,5-4094
1         2
2         4
-----
```

```
D1(config-mst)# revision 2
```

```
D1(config-mst)# instance 1 vlan 3
```

```
D1(config-mst)# instance 2 vlan 5
```

```
D1(config-mst)# show pending
```

```
Pending MST configuration
Name          [CCNPv8]
Revision 2     Instances configured 3
```

```
Instance Vlans mapped
```

```
-----
0         1,6-4094
1         2-3
2         4-5
-----
```

```
D1(config-mst)# exit
```



## Lab - Implement MST

```
D1 (config)# end
```

- j. Now configure D2 and A1 with the same configuration settings (name CCNPv8, revision2, instance 1 vlans 2-3, instance 2 vlans 4-5). After completing the configuration on D2 and A1, the output of **show spanning-tree mst** on A1 should be similar to the following output. D1 in this case is the root for all 3 instances.

```
A1# show spanning-tree mst
```

```
##### MST0    vlans mapped:    1,6-4094
Bridge         address f078.1647.4580  priority      32768 (32768 sysid 0)
Root          address d8b1.9028.af80  priority      32768 (32768 sysid 0)
               port      Fa0/1                path cost     0
Regional Root address d8b1.9028.af80  priority      32768 (32768 sysid 0)
               internal cost 200000    rem hops 19
Operational    hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured     hello time 2 , forward delay 15, max age 20, max hops    20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	200000	128.1	P2p
Fa0/2	Altn	BLK	200000	128.2	P2p
Fa0/3	Altn	BLK	200000	128.3	P2p
Fa0/4	Altn	BLK	200000	128.4	P2p
Fa0/23	Desg	FWD	2000000	128.23	P2p

```
##### MST1    vlans mapped:    2-3
Bridge         address f078.1647.4580  priority      32769 (32768 sysid 1)
Root          address d8b1.9028.af80  priority      32769 (32768 sysid 1)
               port      Fa0/1                cost          200000    rem hops 19
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	200000	128.1	P2p
Fa0/2	Altn	BLK	200000	128.2	P2p
Fa0/3	Altn	BLK	200000	128.3	P2p
Fa0/4	Altn	BLK	200000	128.4	P2p
Fa0/23	Desg	FWD	2000000	128.23	P2p

```
##### MST2    vlans mapped:    4-5
Bridge         address f078.1647.4580  priority      32770 (32768 sysid 2)
Root          address d8b1.9028.af80  priority      32770 (32768 sysid 2)
               port      Fa0/1                cost          200000    rem hops 19
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	200000	128.1	P2p
Fa0/2	Altn	BLK	200000	128.2	P2p
Fa0/3	Altn	BLK	200000	128.3	P2p
Fa0/4	Altn	BLK	200000	128.4	P2p
Fa0/23	Desg	FWD	2000000	128.23	P2p

## Step 2: Controlling the Root Bridge.

Just like with PVST+ and Rapid PVST+, 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.

With PVST+ or Rapid PVST+, root bridge selection is done for each VLAN. With MST, the root bridge is based on instances.

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

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

The difference between the two is the **priority** command will set a specific number (multiple of 4096) as the priority, while 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).

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 mst instance-id 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 increments lower than the default priority) while the **root secondary** command will set the priority on the secondary root to the 28,672 (one increment lower than the default priority).

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

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

- 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.

```
A1# show spanning-tree root
```

MST Instance	Root ID	Root Cost	Hello Time	Max Age	Fwd Dly	Root Port
MST0	24576 d8b1.9028.af80	0	2	20	15	Fa0/1
MST1	24577 d8b1.9028.af80	200000	2	20	15	Fa0/1
MST2	24578 d8b1.905d.c300	200000	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. A1 is the root bridge for MST Instance 0 in this example.

**Step 3: 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 MST. 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 Blocking 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 mst 1
```

```
##### MST1      vlans mapped:    2-3
Bridge          address f078.1647.4580  priority      32769 (32768 sysid 1)
Root           address d8b1.9028.af80  priority      24577 (24576 sysid 1)
               port      Fa0/1          cost          200000      rem hops 19
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	200000	128.1	P2p
Fa0/2	Altn	BLK	200000	128.2	P2p
Fa0/3	Altn	BLK	200000	128.3	P2p
Fa0/4	Altn	BLK	200000	128.4	P2p
Fa0/23	Desg	FWD	2000000	128.23	P2p

```
A1# show spanning-tree blockedports
```

Name	Blocked Interfaces List
MST0	Fa0/2, Fa0/3, Fa0/4
MST1	Fa0/2, Fa0/3, Fa0/4
MST2	Fa0/1, Fa0/2, Fa0/4

```
Number of blocked ports (segments) in the system : 9
```

As you can see, MST 1 has its Root Port on F0/1 and 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 (a value between 1 and 200,000,000) or port priority value (a value between 0 and 240 in increments of 16). 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 of 1000 to F0/2 using the **spanning-tree mst 1 cost value** command, and then issue the **no shutdown** command on 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
```

## Lab - Implement MST

```
A1(config)# interface f0/2
A1(config-if)# spanning-tree mst 1 cost 1000
A1(config-if)# exit
A1(config)# interface range f0/1-2
A1(config-if-range)# no shutdown
A1(config-if-range)# exit
```

- 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 mst 1
```

```
##### MST1      vlans mapped:  2-3
Bridge          address f078.1647.4580  priority      32769 (32768 sysid 1)
Root            address d8b1.9028.af80  priority      24577 (24576 sysid 1)
                port     Fa0/2                cost          1000          rem hops 19
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Altn	BLK	200000	128.1	P2p
Fa0/2	Root	FWD	1000	128.2	P2p
Fa0/3	Desg	FWD	200000	128.3	P2p
Fa0/4	Desg	FWD	200000	128.4	P2p
Fa0/23	Desg	FWD	2000000	128.23	P2p

```
A1# show spanning-tree blockedports
```

Name	Blocked Interfaces List
MST0	Fa0/2, Fa0/3, Fa0/4
MST1	Fa0/1
MST2	Fa0/1, Fa0/2, Fa0/4

```
Number of blocked ports (segments) in the system : 7
```

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 1000.

### Step 4: 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 MST Instance 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 as long as 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 mst 2** and take note of the port ID values listed.

A1# **show spanning-tree mst 2**

```
##### MST2      vlans mapped: 4-5
Bridge          address f078.1647.4580 priority 32770 (32768 sysid 2)
Root            address d8b1.905d.c300 priority 24578 (24576 sysid 2)
                port Fa0/3 cost 200000 rem hops 19
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Altn	BLK	200000	128.1	P2p
Fa0/2	Altn	BLK	200000	128.2	P2p
Fa0/3	Root	FWD	200000	128.3	P2p
Fa0/4	Altn	BLK	200000	128.4	P2p
Fa0/23	Desg	FWD	2000000	128.23	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 by issuing the **spanning-tree mst 2 port-priority value** interface configuration command. Use a value of 64.

D2# **config t**

Enter configuration commands, one per line. End with CNTL/Z.

```
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 mst 2 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
```

- c. On A1, issue the **show spanning-tree mst 2** command and you will see that Fa0/4 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 mst 2**

```
##### MST2      vlans mapped: 4-5
Bridge          address f078.1647.4580 priority 32770 (32768 sysid 2)
Root            address d8b1.905d.c300 priority 24578 (24576 sysid 2)
                port Fa0/4 cost 200000 rem hops 19
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Altn	BLK	200000	128.1	P2p
Fa0/2	Altn	BLK	200000	128.2	P2p
Fa0/3	Altn	BLK	200000	128.3	P2p
Fa0/4	Root	FWD	200000	128.4	P2p

Fa0/23

Desg FWD 2000000 128.23 P2p

### Device Configs - Final

#### Switch D1

```
D1# show run
```

```
Building configuration...
```

```
Current configuration : 9023 bytes
```

```
!  
version 16.9  
no service pad  
service timestamps debug datetime msec  
service timestamps log datetime msec  
! Call-home is enabled by Smart-Licensing.  
service call-home  
no platform punt-keepalive disable-kernel-core  
!  
hostname D1  
!  
!  
vrf definition Mgmt-vrf  
!  
address-family ipv4  
exit-address-family  
!  
address-family ipv6  
exit-address-family  
!  
no aaa new-model  
switch 1 provision ws-c3650-24ts  
!  
login on-success log  
!  
crypto pki trustpoint SLA-TrustPoint  
enrollment pkcs12  
revocation-check crl  
!  
license boot level ipservicesk9  
!  
!  
diagnostic bootup level minimal  
!  
spanning-tree mode mst  
spanning-tree extend system-id  
!  
spanning-tree mst configuration  
name CCNPv8  
revision 2
```

## Lab - Implement MST

---

```
instance 1 vlan 2-3
instance 2 vlan 4-5
!
spanning-tree mst 0-1 priority 24576
spanning-tree mst 2 priority 28672
!
!
redundancy
mode sso
!
transceiver type all
monitoring
!
class-map match-any system-cpp-police-topology-control
description Topology control
class-map match-any system-cpp-police-sw-forward
description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL
SAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
description L2 LVX control packets
class-map match-any system-cpp-police-forus
description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
description MCAST END STATION
class-map match-any system-cpp-police-multicast
description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
description L2 control
class-map match-any system-cpp-police-dot1x-auth
description DOT1X Auth
class-map match-any system-cpp-police-data
description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
description DHCP snooping
class-map match-any system-cpp-police-system-critical
description System Critical and Gold Pkt
!
```

## Lab - Implement MST

---

```
policy-map system-cpp-policy
!
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 switchport mode trunk
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 switchport mode trunk
!
interface GigabitEthernet1/0/6
 switchport mode trunk
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 shutdown
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
```



## Lab - Implement MST

---

```
!  
interface GigabitEthernet1/0/15  
 shutdown  
!  
interface GigabitEthernet1/0/16  
 shutdown  
!  
interface GigabitEthernet1/0/17  
 shutdown  
!  
interface GigabitEthernet1/0/18  
 shutdown  
!  
interface GigabitEthernet1/0/19  
 shutdown  
!  
interface GigabitEthernet1/0/20  
 shutdown  
!  
interface GigabitEthernet1/0/21  
 shutdown  
!  
interface GigabitEthernet1/0/22  
 shutdown  
!  
interface GigabitEthernet1/0/23  
 shutdown  
!  
interface GigabitEthernet1/0/24  
 shutdown  
!  
interface GigabitEthernet1/1/1  
 shutdown  
!  
interface GigabitEthernet1/1/2  
 shutdown  
!  
interface GigabitEthernet1/1/3  
 shutdown  
!  
interface GigabitEthernet1/1/4  
 shutdown  
!  
interface Vlan1  
 no ip address  
 shutdown  
!  
ip forward-protocol nd  
ip http server  
ip http authentication local
```

## Lab - Implement MST

---

```
ip http secure-server
!
!
control-plane
service-policy input system-cpp-policy
!
banner motd ^C D1, Multiple Spanning Tree ^C
!
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line aux 0
stopbits 1
line vty 0 4
exec-timeout 0 0
privilege level 15
password cisco123
logging synchronous
login
line vty 5 15
login
!
!
end
```

### Switch D2

```
D2# show run
Building configuration...

Current configuration : 8981 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D2
!
vrf definition Mgmt-vrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
```

## Lab - Implement MST

---

```
!  
no aaa new-model  
switch 1 provision ws-c3650-24ts  
!  
login on-success log  
!  
license boot level ipservicesk9  
!  
diagnostic bootup level minimal  
!  
spanning-tree mode mst  
spanning-tree extend system-id  
!  
spanning-tree mst configuration  
  name CCNPv8  
  revision 2  
  instance 1 vlan 2-3  
  instance 2 vlan 4-5  
!  
spanning-tree mst 1 priority 28672  
spanning-tree mst 2 priority 24576  
!  
redundancy  
  mode sso  
!  
transceiver type all  
  monitoring  
!  
class-map match-any system-cpp-police-topology-control  
  description Topology control  
class-map match-any system-cpp-police-sw-forward  
  description Sw forwarding, L2 LVX data, LOGGING  
class-map match-any system-cpp-default  
  description Inter FED, EWLC control, EWLC data  
class-map match-any system-cpp-police-sys-data  
  description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL  
  SAMPLED DATA, RPF Failed  
class-map match-any system-cpp-police-punt-webauth  
  description Punt Webauth  
class-map match-any system-cpp-police-l2lvx-control  
  description L2 LVX control packets  
class-map match-any system-cpp-police-forus  
  description Forus Address resolution and Forus traffic  
class-map match-any system-cpp-police-multicast-end-station  
  description MCAST END STATION  
class-map match-any system-cpp-police-multicast  
  description Transit Traffic and MCAST Data  
class-map match-any system-cpp-police-l2-control  
  description L2 control  
class-map match-any system-cpp-police-dot1x-auth
```

## Lab - Implement MST

---

```
description DOT1X Auth
class-map match-any system-cpp-police-data
description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
description DHCP snooping
class-map match-any system-cpp-police-system-critical
description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface GigabitEthernet0/0
vrf forwarding Mgmt-vrf
no ip address
shutdown
negotiation auto
!
interface GigabitEthernet1/0/1
switchport mode trunk
!
interface GigabitEthernet1/0/2
shutdown
!
interface GigabitEthernet1/0/3
shutdown
!
interface GigabitEthernet1/0/4
shutdown
!
interface GigabitEthernet1/0/5
switchport mode trunk
!
interface GigabitEthernet1/0/6
switchport mode trunk
spanning-tree mst 2 port-priority 64
!
interface GigabitEthernet1/0/7
shutdown
!
interface GigabitEthernet1/0/8
shutdown
!
interface GigabitEthernet1/0/9
shutdown
```

## Lab - Implement MST

---

```
!  
interface GigabitEthernet1/0/10  
 shutdown  
!  
interface GigabitEthernet1/0/11  
 shutdown  
!  
interface GigabitEthernet1/0/12  
 shutdown  
!  
interface GigabitEthernet1/0/13  
 shutdown  
!  
interface GigabitEthernet1/0/14  
 shutdown  
!  
interface GigabitEthernet1/0/15  
 shutdown  
!  
interface GigabitEthernet1/0/16  
 shutdown  
!  
interface GigabitEthernet1/0/17  
 shutdown  
!  
interface GigabitEthernet1/0/18  
 shutdown  
!  
interface GigabitEthernet1/0/19  
 shutdown  
!  
interface GigabitEthernet1/0/20  
 shutdown  
!  
interface GigabitEthernet1/0/21  
 shutdown  
!  
interface GigabitEthernet1/0/22  
 shutdown  
!  
interface GigabitEthernet1/0/23  
 shutdown  
!  
interface GigabitEthernet1/0/24  
 shutdown  
!  
interface GigabitEthernet1/1/1  
 shutdown  
!  
interface GigabitEthernet1/1/2
```

## Lab - Implement MST

---

```
shutdown
!  
interface GigabitEthernet1/1/3  
shutdown  
!  
interface GigabitEthernet1/1/4  
shutdown  
!  
interface Vlan1  
no ip address  
shutdown  
!  
ip forward-protocol nd  
ip http server  
ip http authentication local  
ip http secure-server  
!  
control-plane  
service-policy input system-cpp-policy  
!  
banner motd ^C D2, Multiple Spanning Tree ^C  
!  
line con 0  
exec-timeout 0 0  
logging synchronous  
stopbits 1  
line aux 0  
stopbits 1  
line vty 0 4  
login  
line vty 5 15  
login  
end
```

### Switch A1

```
A1# show run  
Building configuration...  
  
Current configuration : 2095 bytes  
!  
version 15.2  
no service pad  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname A1  
!  
boot-start-marker
```

## Lab - Implement MST

---

```
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
!
!
spanning-tree mode mst
spanning-tree extend system-id
!
spanning-tree mst configuration
 name CCNPv8
 revision 2
 instance 1 vlan 2-3
 instance 2 vlan 4-5
!
vlan internal allocation policy ascending
!
interface FastEthernet0/1
 switchport mode trunk
!
interface FastEthernet0/2
 switchport mode trunk
 spanning-tree mst 1 cost 1000
!
interface FastEthernet0/3
 switchport mode trunk
!
interface FastEthernet0/4
 switchport mode trunk
!
interface FastEthernet0/5
 switchport mode trunk
!
interface FastEthernet0/6
 switchport mode trunk
!
interface FastEthernet0/7
 switchport mode trunk
!
interface FastEthernet0/8
 switchport mode trunk
!
interface FastEthernet0/9
 switchport mode trunk
!
interface FastEthernet0/10
 switchport mode trunk
!
interface FastEthernet0/11
```

## Lab - Implement MST

---

```
switchport mode trunk
!  
interface FastEthernet0/12  
switchport mode trunk  
!  
interface FastEthernet0/13  
switchport mode trunk  
!  
interface FastEthernet0/14  
switchport mode trunk  
!  
interface FastEthernet0/15  
switchport mode trunk  
!  
interface FastEthernet0/16  
switchport mode trunk  
!  
interface FastEthernet0/17  
switchport mode trunk  
!  
interface FastEthernet0/18  
switchport mode trunk  
!  
interface FastEthernet0/19  
switchport mode trunk  
!  
interface FastEthernet0/20  
switchport mode trunk  
!  
interface FastEthernet0/21  
switchport mode trunk  
!  
interface FastEthernet0/22  
switchport mode trunk  
!  
interface FastEthernet0/23  
switchport mode trunk  
!  
interface FastEthernet0/24  
shutdown  
!  
interface GigabitEthernet0/1  
switchport mode trunk  
!  
interface GigabitEthernet0/2  
switchport mode trunk  
!  
interface Vlan1  
!  
ip http server
```



## Lab - Implement MST

---

```
ip http secure-server
!  
banner motd ^C A1, Multiple Spanning Tree ^C  
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
line vty 0 4  
  login  
line vty 5 15  
  login  
!  
end
```