Chapter 4: **Spanning Tree** in Depth

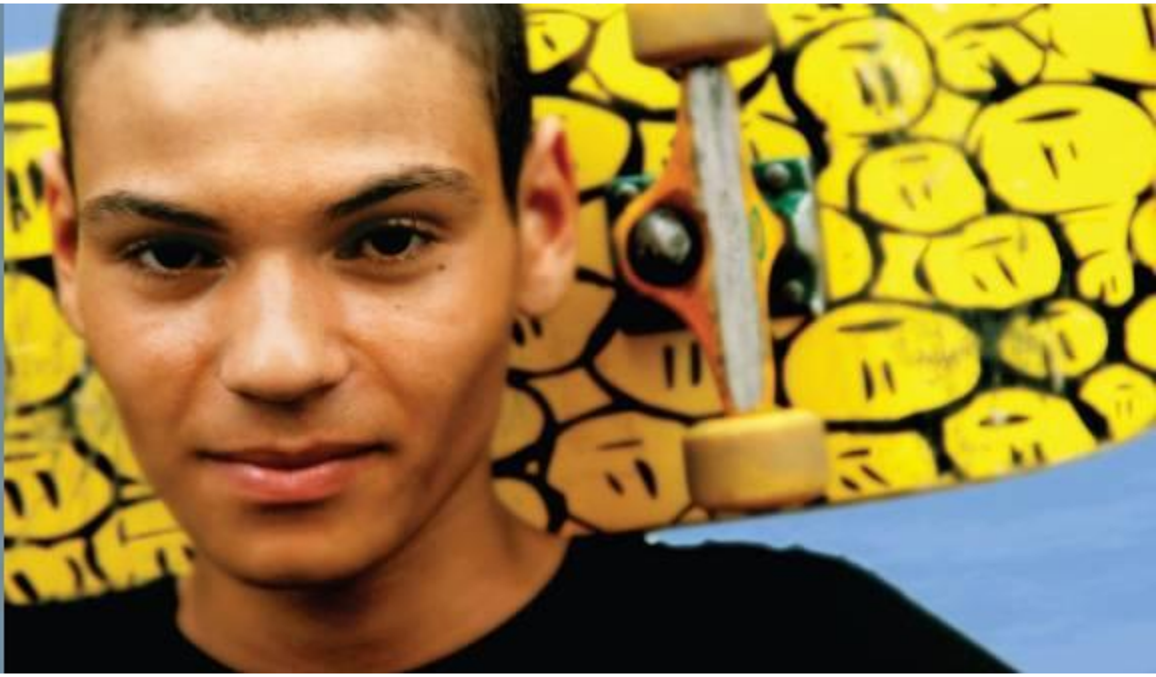
CCNP SWITCH: Implementing Cisco IP Switched Networks

Cisco | Networking Academy®
Mind Wide Open™

Chapter 4 Objectives

- Spanning Tree Protocol (STP) overview, its operations, and history
- Implement Rapid Spanning Tree Protocol (RSTP)
- Describe how and where to configure the following features: PortFast, UplinkFast, BackboneFast, BPDU Guard, BPDU Filter, Root Guard, Loop Guard, Unidirectional Link Detection, and FlexLinks
- Configure Multiple Spanning Tree (MST)
- Troubleshooting STP

Spanning Tree Protocol (STP) Overview



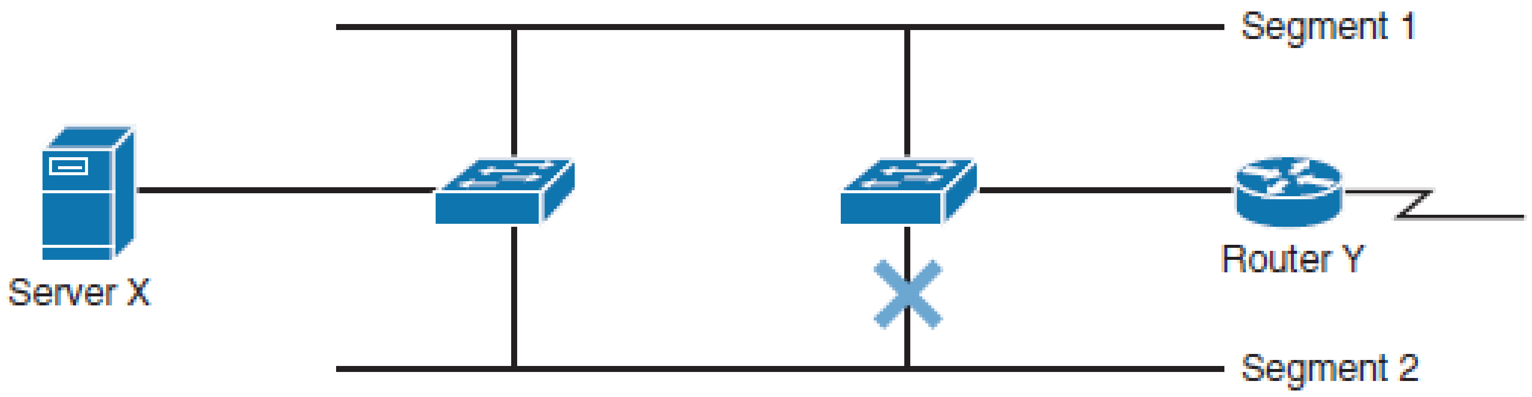
Spanning Tree Protocol Overview

Upon completing this section, you will be able to meet these objectives:

- Explain the need for STP
- List different standards of STP
- Describe basic STP operation
- Describe bridge protocol data units (BPDU)
- Explain the root bridge election
- Explain the root port election
- Explain designated port election
- Explain STP port states
- Explain PVST+
- Explain STP topology changes

STP Need

- Redundant topology can eliminate single points of failure in the network , however, STP blocks certain ports, so there is only one active path to each segment .



Redundant Topology Problems

- **Broadcast storms**
  - Each switch on a redundant network floods broadcast frames endlessly. These frames then travel around the loop in all directions.
- **Multiple frame transmission**
  - Multiple copies of the same unicast frames may be delivered to destination station, which can cause problems with the receiving protocol. Multiple copies of the same frame can cause unrecoverable errors.
- **MAC database instability**
  - If a loop occurs, the same source MAC address could be seen on multiple interfaces causing instability. Data forwarding can be impaired when the switch consumes the resources that are coping with instability in the MAC address table.

Solution

- STP allows physical path redundancy while preventing the undesirable effects of active loops in the network.
- STP forces certain ports into a standby state so that they do not listen, forward, or flood data frames.
- There is only one active path to each network segment.
- If there is a problem with connectivity to any of the segments, STP reestablishes connectivity by automatically activating a previously inactive path.

Solution

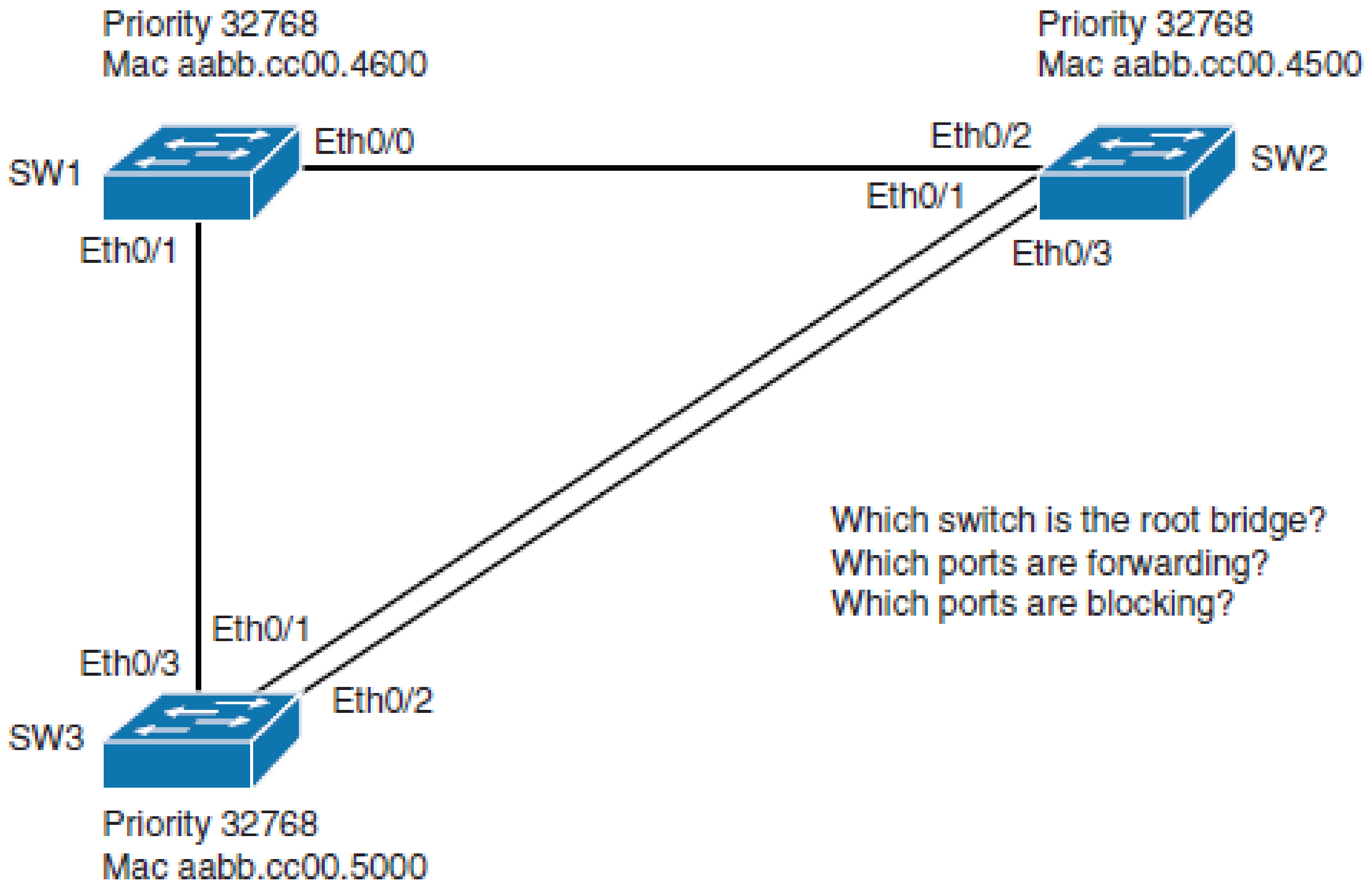
- STP uses bridge protocol data units (BPDUs) for its operations.
- BPDUs are messages that STP uses to determine current topology information and how to react if any devices added or removed or changed in the topology.
- By default, they are sent out every 2 seconds on all switch ports.
- BDPUs are discussed later in detail

STP Standards

	Standard	Resources Needed	Convergence	VLAN
STP	Original 802.1D	Low	Slow	One
CST	IEEE 802.1D	Low	Slow	All
PVST+	Cisco	High	Slow	Per vlan
RSTP	IEEE 802.1w	Medium	Fast	All
RPVST+	Cisco	Very high	Fast	Per vlan
MST	IEEE 802.1s	Medium or high	Fast	Vlan list

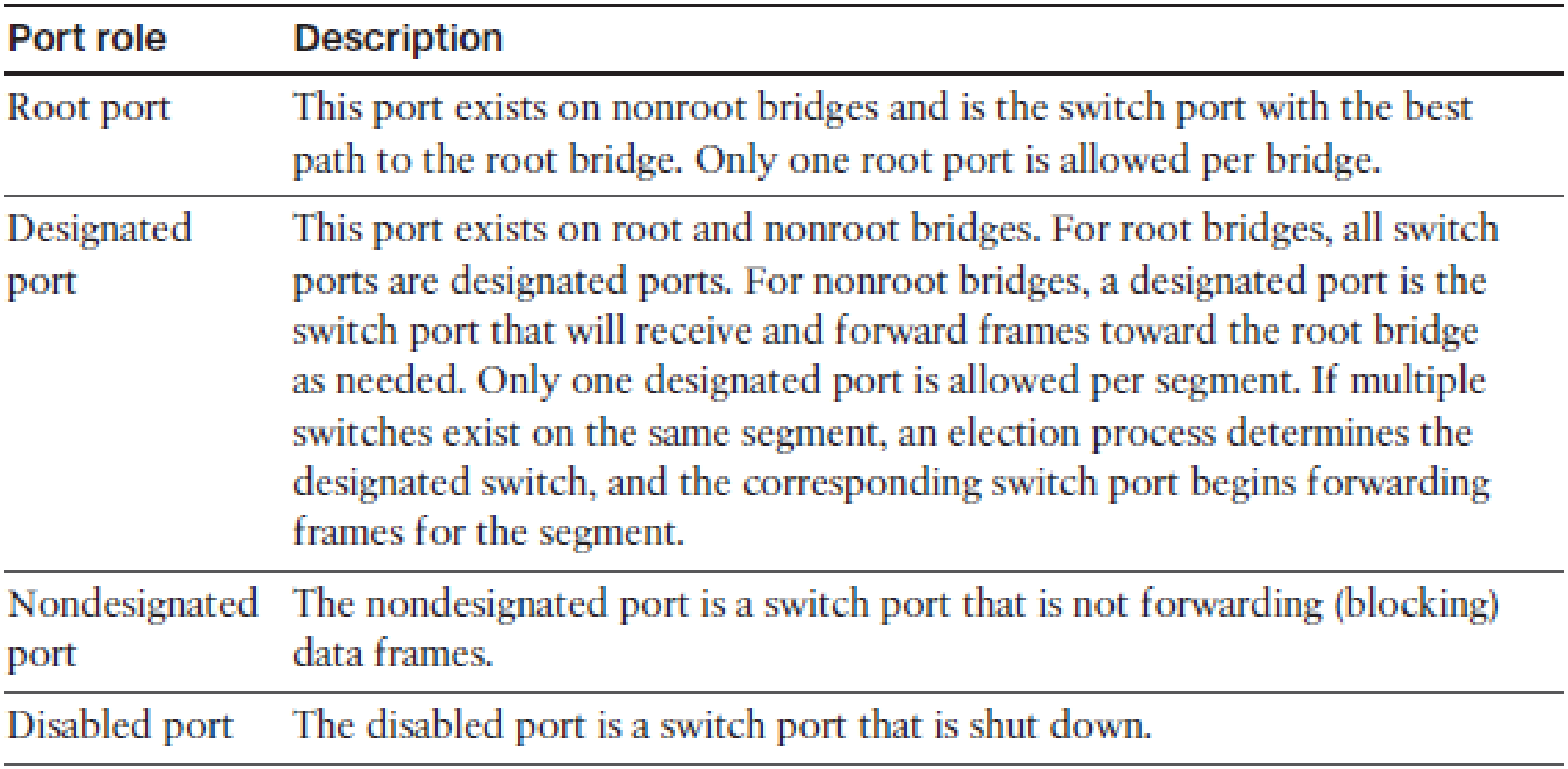
- Common Spanning Tree (CST) assumes one spanning-tree instance for the entire network
- By default, Cisco switches use PVST+ (Per-VLAN Spanning Tree Protocol Plus).
- However, it is highly recommended to use either Rapid PVST+ (RPVST+) or MST (Multiple Spanning Tree) wherever possible

STP Operations



Which switch is the root bridge?
Which ports are forwarding?
Which ports are blocking?

STP Port Roles



Port role	Description
Root port	This port exists on nonroot bridges and is the switch port with the best path to the root bridge. Only one root port is allowed per bridge.
Designated port	This port exists on root and nonroot bridges. For root bridges, all switch ports are designated ports. For nonroot bridges, a designated port is the switch port that will receive and forward frames toward the root bridge as needed. Only one designated port is allowed per segment. If multiple switches exist on the same segment, an election process determines the designated switch, and the corresponding switch port begins forwarding frames for the segment.
Nondesignated port	The nondesignated port is a switch port that is not forwarding (blocking) data frames.
Disabled port	The disabled port is a switch port that is shut down.

STP Operations

STP provides loop resolution by managing the physical path to the given network segment, by performing the following three steps:

1. Elects one root bridge

- Only one bridge can act as the root bridge. The root bridge is the reference point; all data flows in the network are from the perspective of this switch. All ports on a root bridge are forwarding traffic.

2. Selects the root port on the non-root bridge

- One port on each non-root bridge is the root port. It is the port with the lowest-cost path from the non-root bridge to the root bridge. By default, STP path cost is calculated from the bandwidth of the link. You can also set STP path cost manually.

3. Selects the designated port on each segment

- There is one designated port on each segment. It is selected on the bridge with the lowest-cost path to the root bridge.

Bridge Protocol Data Units

- STP uses **BPDUs** to exchange STP information, specifically for root bridge election and for loop identification.
- By default, BPDUs are sent out every 2 seconds.
- BPDUs are generally categorized into two types:
  - **Configuration BDPUs**
    - Used for calculating the STP
  - **TCN (Topology Change Notification) BPDUs**
    - Used to inform changes in the network topology

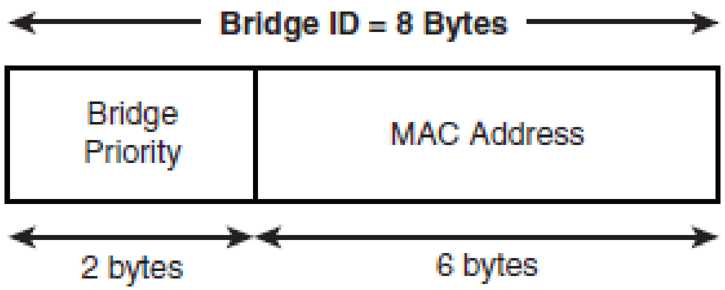
The BPDU Frame

Protocol ID	Version	Message Type	Flags	Root Bridge ID	Root Path Cost	Sender Bridge ID	Port ID	Massage Age	Maximum Age	Hello Time	Forward Delay

- **Protocol ID:** Identifies that this is an STP frame (0x0000)
- **Version:** Identifies the current version of the protocol (0x00 - STP, 0x02 - RSTP)
- **Message Type:** Identifies the type of BPDU (configuration or TCN BPDU)
- **Flags:** Used in response to a TCN BPDU (0 – no change, 1 - change)
- **Root Bridge ID:** Identifies the bridge ID (BID) of the root bridge
- **Root Path Cost:** Identifies the cost from the transmitting switch to the root
- **Sender Bridge ID:** Identifies the BID of the transmitting switch
- **Port ID:** Identifies the transmitting port (Priority + port ID)
- **Message Age:** Indicates the age of the current BPDU
- **Max Age:** Indicates the timeout value
- **Hello Time:** Identifies the time interval between generations of configuration BPDUs by the root
- **Forward Delay:** Defines the time a switch port must wait in the listening and learning state
- **Destination MAC address 01-80-c2-00-00-00 and source MAC is port MAC**

Root Bridge Election

- The root bridge is chosen with an election.
- In STP, each switch has a unique bridge Identifier (BID) that consists of the following:
  - Bridge priority (a value between 0 and 65 535, with the default being 32 768)
  - MAC address



  - newer STP versions added Extended system ID = **VLAN ID**
- The root bridge is selected based on the lowest BID.
- If all switches in the network have the same priority, the switch with the lowest MAC address becomes the root bridge

Root Bridge Election

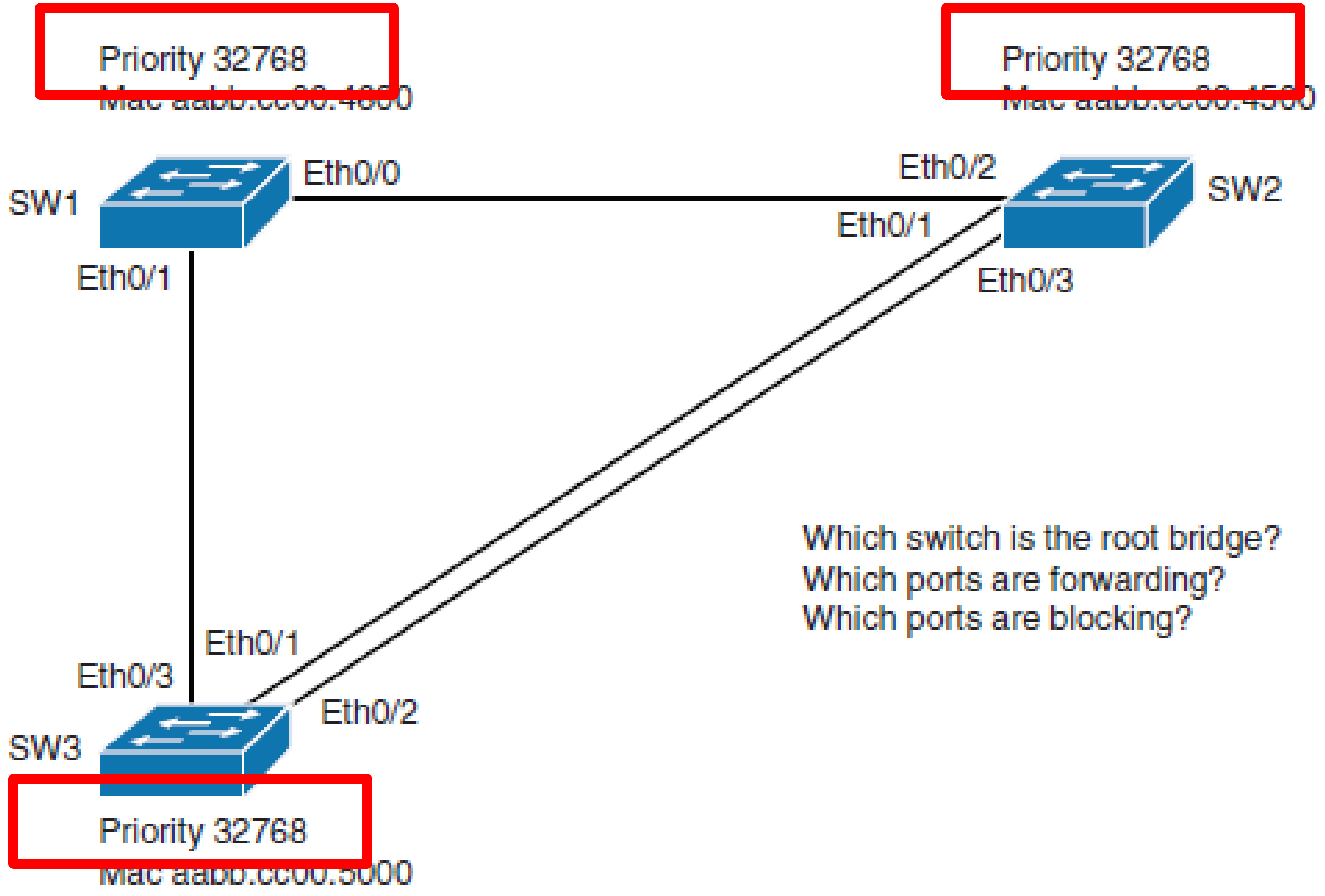
- The root bridge is the logical center of the spanning tree topology
- In the beginning, each switch assumes that it is the root bridge. Each switch sends BPDUs to its neighbours, presenting its BID. At the same time, it receives the BPDUs from all the neighbours.
- Each time a switch receives a BPDU, it checks its BID against its own. If the received BID is higher than its own, then the switch knows it is not the root bridge. converges, it keeps its assumption of being the root bridge.
- Eventually the election converges, and all switches agree that one of them is the root bridge.
- Root bridge election is an ongoing process. So, if a new switch appears with a better BID, it will be elected as the new root bridge.

Root Port Election

- After the root bridge is elected, each non-root bridge must figure out where it is in relation to the root bridge.
- Root port is the port with the best path to the root bridge.
- To determine root ports on non-root bridges, cost value is used.
- The path cost is the **cumulative cost** of all links to the root bridge.
- Root port indicates the lowest cost to the root bridge.

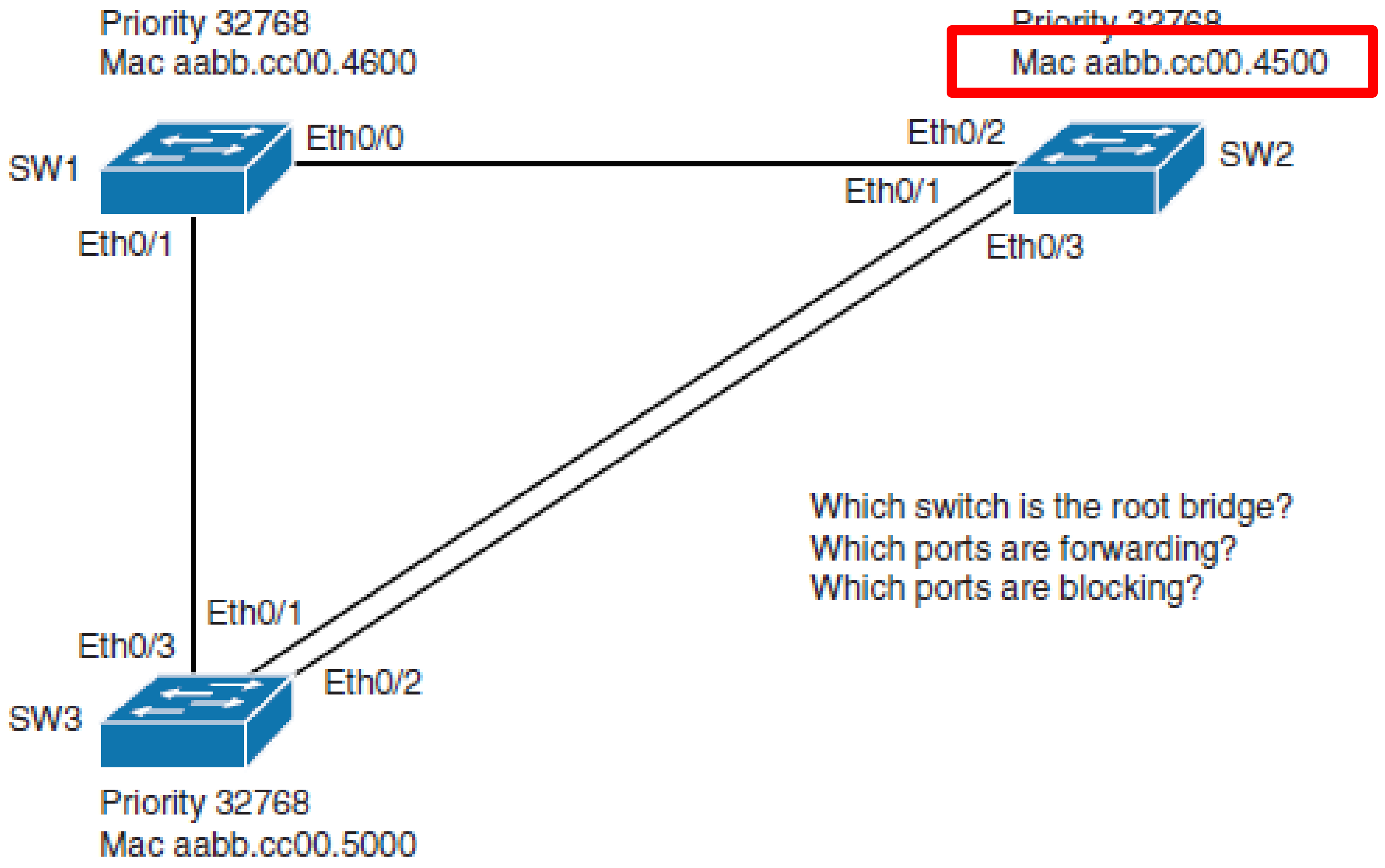
Link Speed	**Correct IEEE 802.1D-2004 Cost**	**Correct IEEE 802.1D-1998 Cost**
10 Gbps	2	1
1 Gbps	4	1
100 Mbps	19	10
10 Mbps	100	100

STP Operations



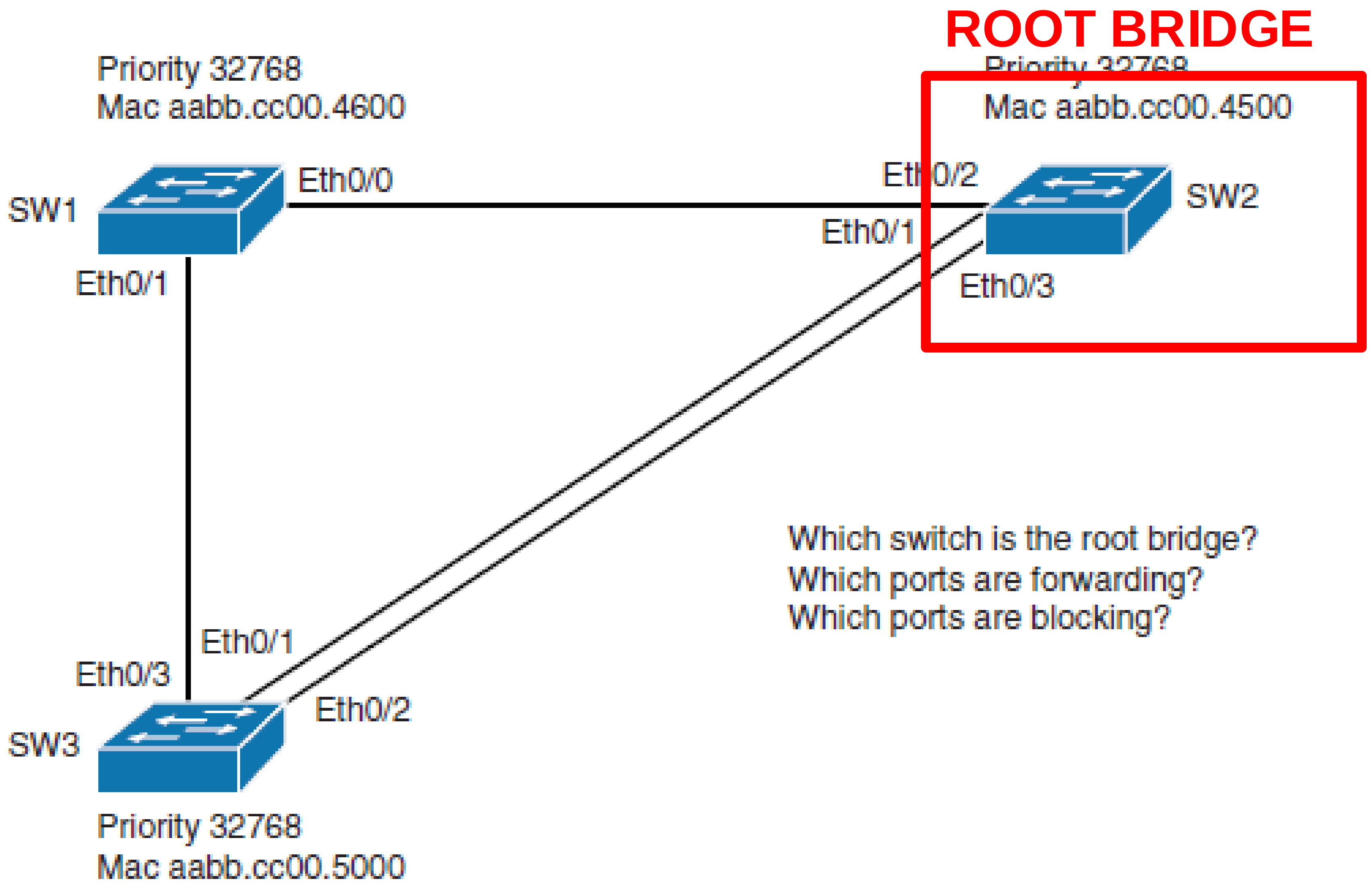
Which switch is the root bridge?
Which ports are forwarding?
Which ports are blocking?

STP Operations



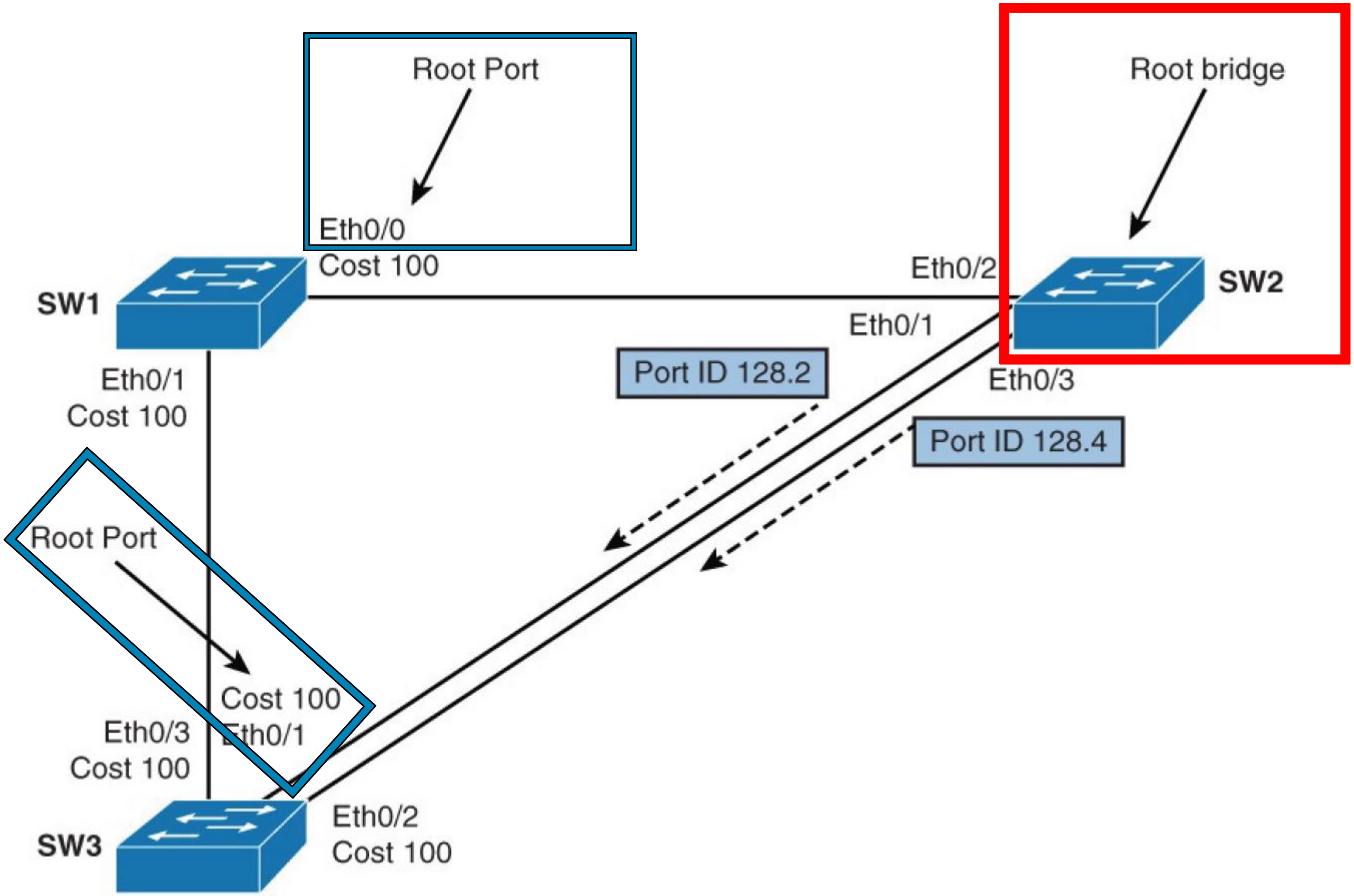
Which switch is the root bridge?
Which ports are forwarding?
Which ports are blocking?

STP Operations



Which switch is the root bridge?
Which ports are forwarding?
Which ports are blocking?

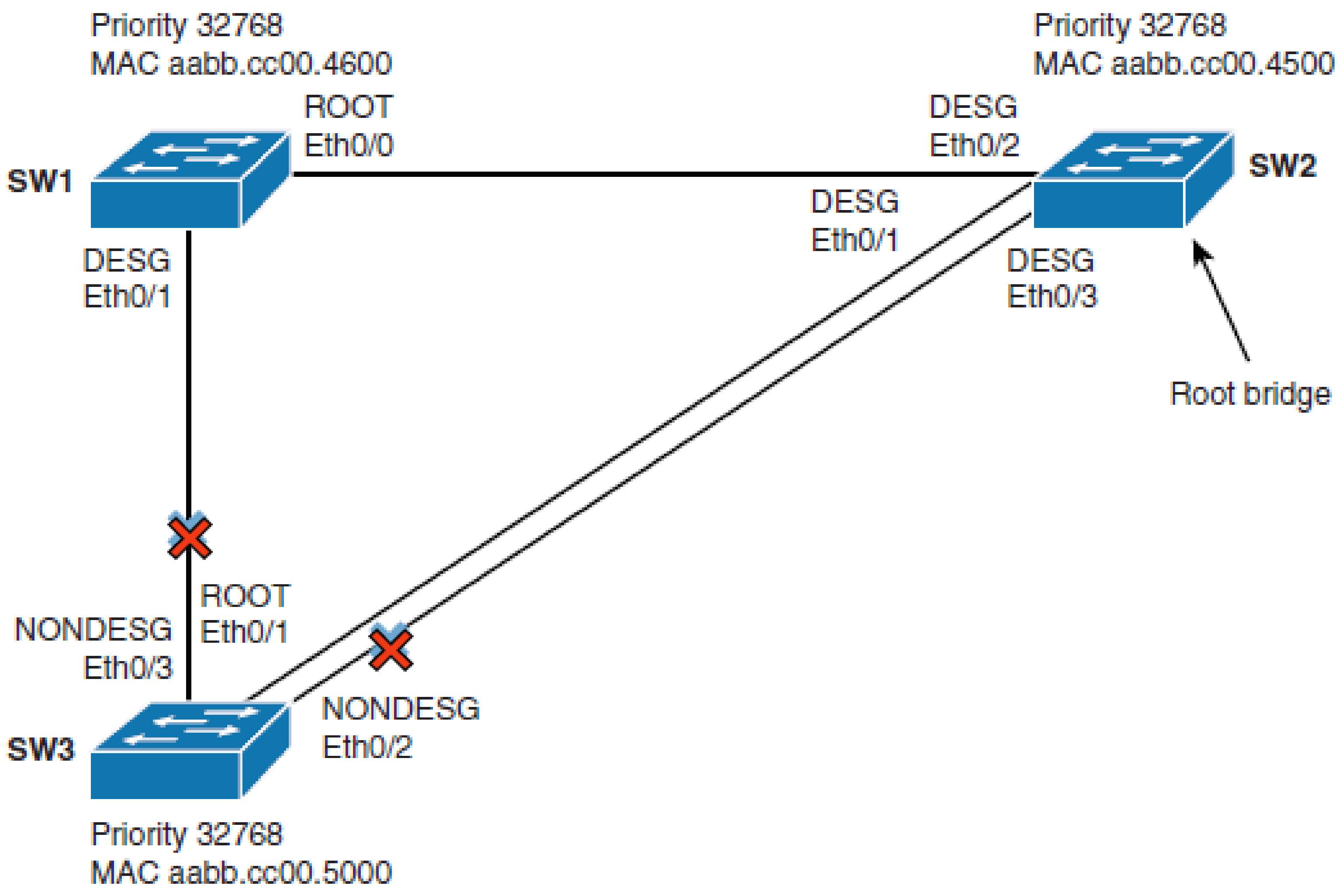
Root Port Election



Designated Port Election

- After the root bridge and root ports on non-root bridges have been elected, to prevent the loops STP must identify which port on the segment will forward the traffic.
- Only one of the links on a segment should forward traffic to and from that segment.
- The designated port, the one forwarding the traffic, is also chosen based on the lowest cost to the root bridge.
- On the root bridge, all ports are designated.
- If there are two paths with equal cost to the root bridge, STP uses the following criteria for best path determination and consequently for determination of designated and non-designated ports on the segment:
  - Lowest BID
  - Lowest sender BID
  - Lowest sender port ID

STP Process


Priority 32768
MAC aabb.cc00.4600
ROOT
Eth0/0
SW1
DESG
Eth0/1
Priority 32768
MAC aabb.cc00.4500
DESG
Eth0/2
SW2
DESG
Eth0/1
DESG
Eth0/3
Root bridge
ROOT
Eth0/1
NONDESG
Eth0/3
SW3
NONDESG
Eth0/2
Priority 32768
MAC aabb.cc00.5000

STP Port States

Blocking (`Sts = BLK`)

- In this state, the port ensures that no bridging loops occur.
- A port in this state cannot receive or transmit data, but it receives BPDUs, so the switch can hear from its neighbour switches and determine the location, and root bridge ID, of the root switch and port roles of each switch.
- A port in this state is a **non-designated port**, and therefore it does not participate in active topology.

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------
Fa0/7               Altn BLK 19        128.7    P2p
Fa0/8               Root FWD 19        128.8    P2p
```

STP Port States

Listening (`Sts = LIS`)

- A port is moved from blocking to listening state if there is a possibility to be selected as the root or designated port.
- The port in this state still cannot send or receive data frames.
- But it is **allowed to send** and **receive** BPDUs, so it is participating in active topology.

Learning (`Sts = LRN`)

- After a period of time (forward delay) in listening state, the port is moved to learning state.
- The port still sends and receives BPDUs; in addition, it can learn and add new MAC addresses to its table.
- A port in this state still cannot send any data frames.

STP Port States

Forwarding (`Sts = FWD`)

- After another period of time (forward delay) in learning state, the port is moved to forwarding state.
- It is considered as part of the active topology. It sends and receives data frames and also sends and receives BPDUs.

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------
Fa0/8               Root FWD 19        128.8    P2p
```

Disabled

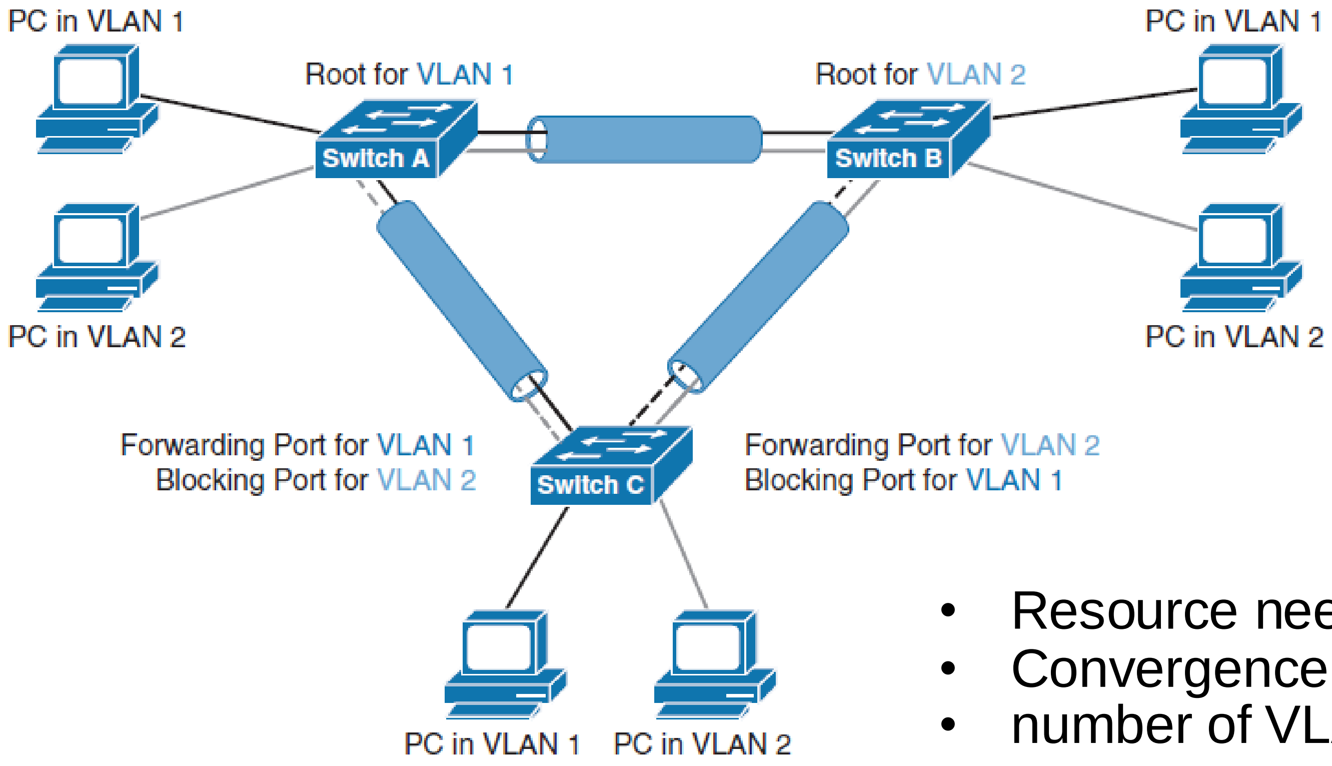
- In this state, a port is administratively shut down.
- It does not participate in spanning tree, and it does not forward frames.

STP Port States

STP Port State	Receive BPDUs	Send BPDUs	Learn MAC Addresses	Receive Data	Send Data	Duration of State
Blocking	✓ Yes	✗ No	✗ No	✗ No	✗ No	Undefined (if there is a loop)
Listening	✓ Yes	✓ Yes	✗ No	✗ No	✗ No	Forward delay (15 seconds)
Learning	✓ Yes	✓ Yes	✓ Yes	✗ No	✗ No	Forward delay (15 seconds)
Forwarding	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	Undefined (as long as there is no loop)
Disabled	✗ No	✗ No	✗ No	✗ No	✗ No	Until administrator enables it

Per-VLAN STP Plus (PVST+)

- Per-VLAN STP Plus (PVST+) is a Cisco implementation of STP that provides a separate spanning-tree instance for each configured VLAN in the network.

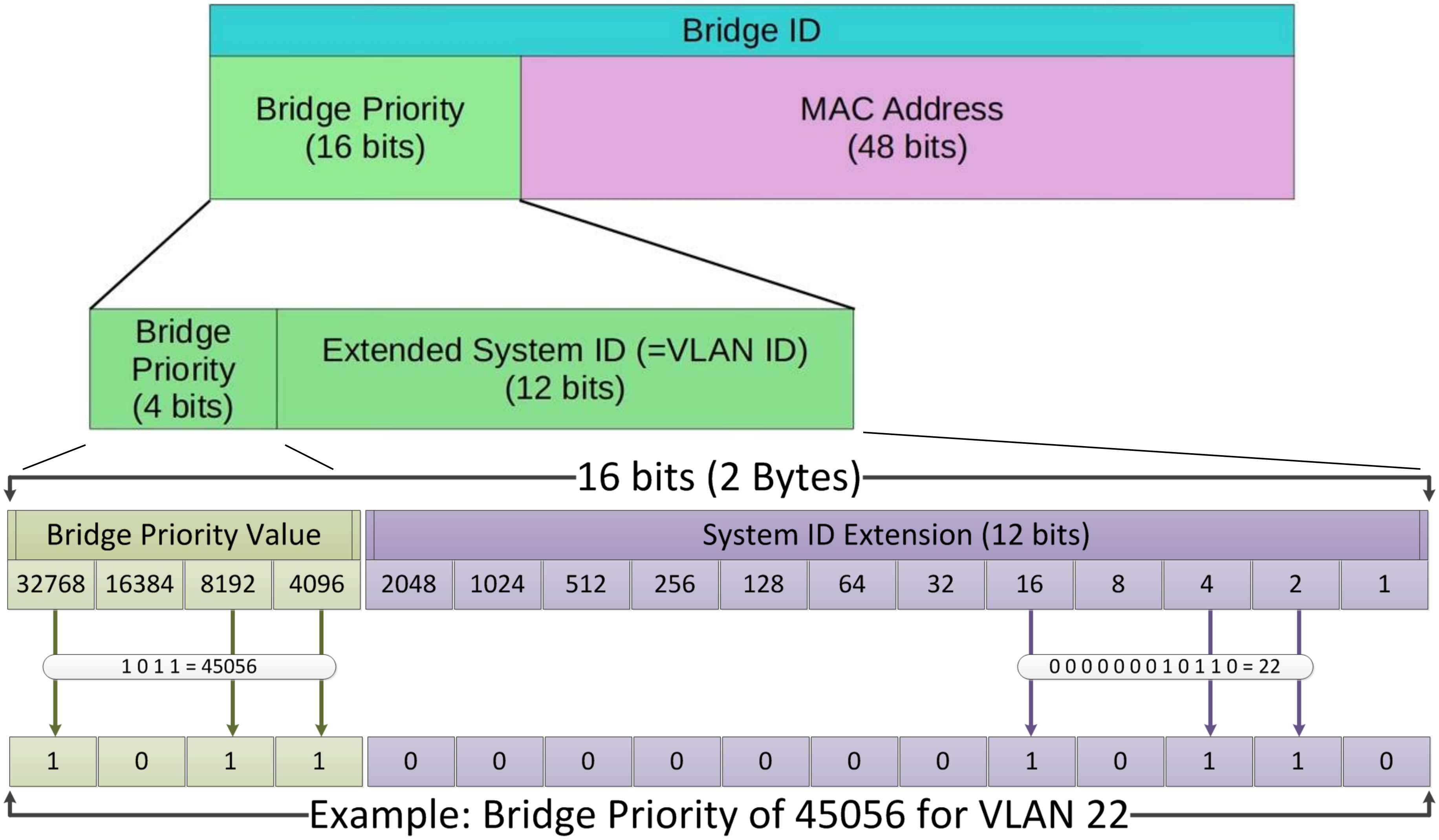


• Resource needed:	**High**
• Convergence:	**Slow**
• number of VLANs:	**per VLAN**

Per-VLAN STP Plus (PVST+)

- Spanning-tree operation requires that each switch has a unique BID per VLAN instance.
- To carry BID information, the extended system ID is accommodated.
- The original 16-bit bridge priority field is split into two fields, resulting in the following components in the BID:
  - **Bridge priority**
    - A 4-bit field used to carry bridge priority.
    - The default priority is 32 768, which is the midrange value.
    - The priority is conveyed in discrete values in increments of **4096**.
  - **Extended system ID**
    - A 12-bit field carrying the VLAN ID

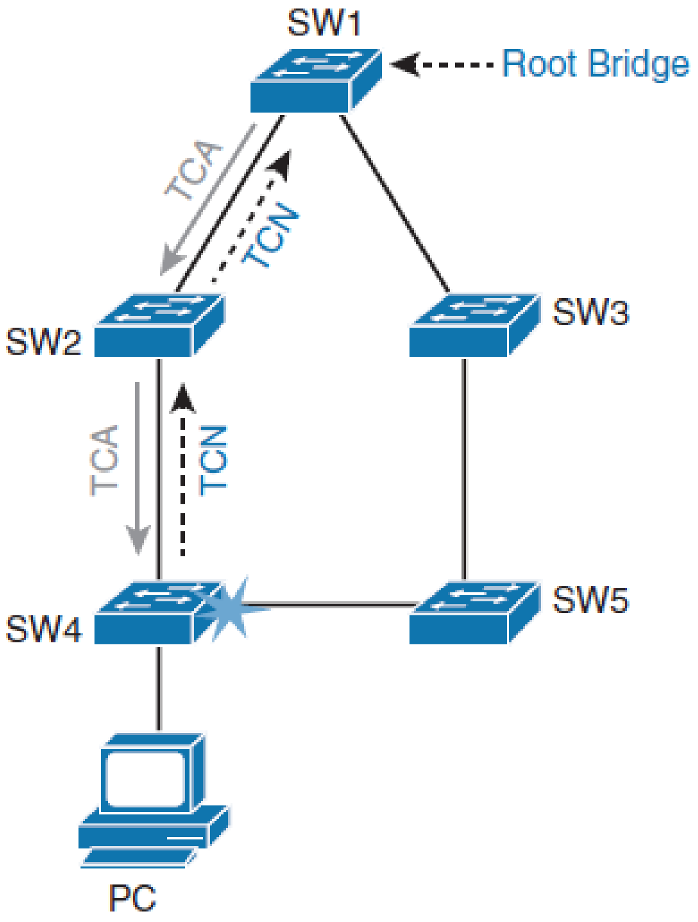
Per-VLAN STP Plus (PVST+)



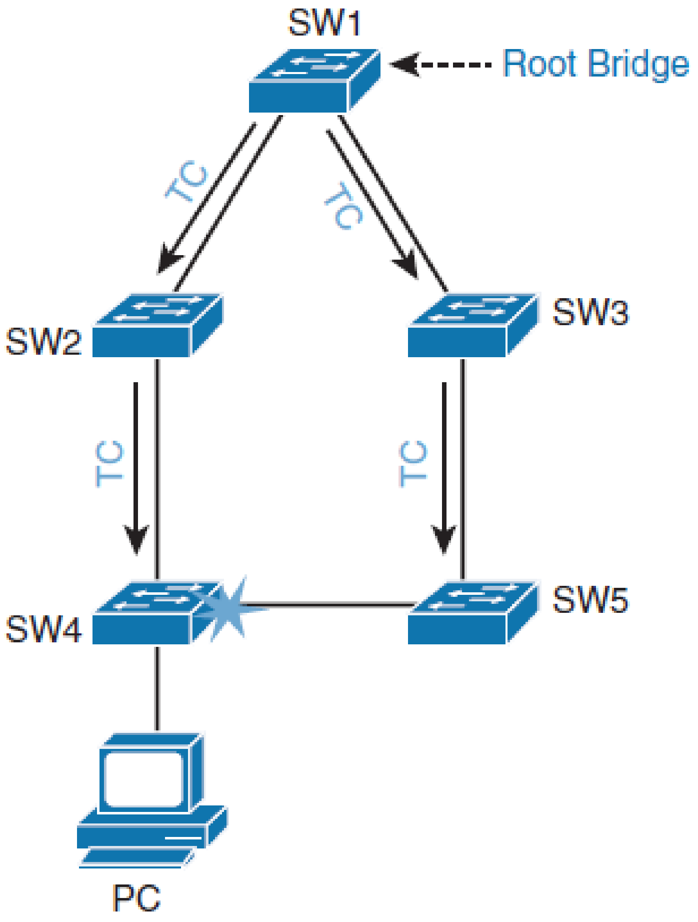
STP Topology Changes

Topology change = switchport changes state to Forwarding or Blocking.

Switch announce topology change to the root bridge.



Root bridge signals the topology change to other switches.



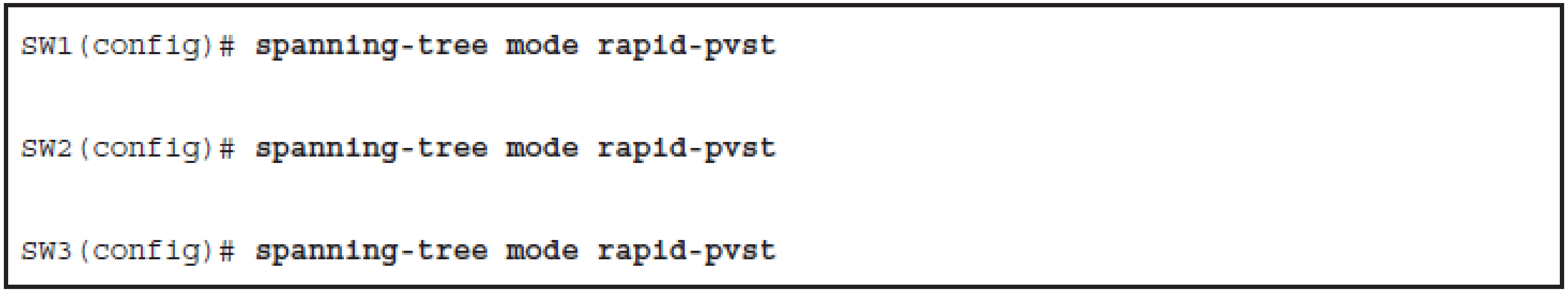
Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol

Upon completing this section, you will be able to meet these objectives:

- List and explain RSTP port roles
- Compare RSTP and STP port states
- Explain how STP handles topology changes
- Describe RSTP link types
- Configure and modify STP behaviour
- Explain how RSTP handles topology changes

Changing the STP Mode to RSTP



```
SW1(config)# spanning-tree mode rapid-pvst

SW2(config)# spanning-tree mode rapid-pvst

SW3(config)# spanning-tree mode rapid-pvst
```

- The convergence time for RSTP is much shorter than for STP. The entire convergence happens at the speed of BPDU transmission.
- That can be less than 1 second.

RSTP Overview

- Rapid Spanning Tree Protocol (RSTP) is an enhancement of the original Spanning Tree Protocol (STP) that significantly improves convergence time, network efficiency, and fault tolerance in Ethernet networks. It addresses the slow response times of traditional STP while maintaining compatibility with older implementations.

1. **Faster Convergence**
2. **Port Roles and**
3. **Edge Ports (**Similar to PortFast in STP**)**
4. **Backward Compatibility**
5. **Prevention of Temporary Loops**
6. **Automatic Detection of Topology Changes**
7. **Multiple Spanning Tree (**MST 802.1s**) Compatibility**

RSTP Faster Convergence

- One of the major drawbacks of the original STP (802.1D) was its slow convergence time, taking anywhere between 30 to 50 seconds to respond to topology changes.
- RSTP overcomes this limitation by implementing mechanisms that allow for near-instantaneous convergence, typically within 1 to 3 seconds.
- Traditional STP requires a **Listening (15s)** → **Learning (15s)** → **Forwarding** process.
- RSTP eliminates these delays by moving ports quickly to a forwarding state using new roles and a handshake mechanism – Proposal/Agreement mechanism

RSTP Port Roles

- RSTP significantly speeds the recalculation of the spanning tree when the network topology changes. Port roles:
- **Root port (`Role = Root`)**
  - The root port is the switch port on every non-root bridge that is the chosen path to the root bridge.
  - There can be only (has to be) one root port on every switch.
  - The root port is considered as part of active topology.
  - It forwards, sends, and receives BPDUs and data messages.
- **Designated port (`Role = Desg`)**
  - Each switch has at least one switch port as the designated port for the segment.
  - In active topology, the switch with the designated port receives frames on the segment that are destined for the root bridge.
  - There can be only one designated port per segment.

RSTP Port Roles

- **Alternate (`Role = Altn`)**
  - The alternate port is a switch port that offers an alternate path toward the root bridge.
  - It assumes a discarding state (**BLK**) in an active topology.
  - The alternate port makes a transition to a root port if the current root path (port) fails.

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------
Fa0/7               Altn BLK 19        128.7    P2p
```

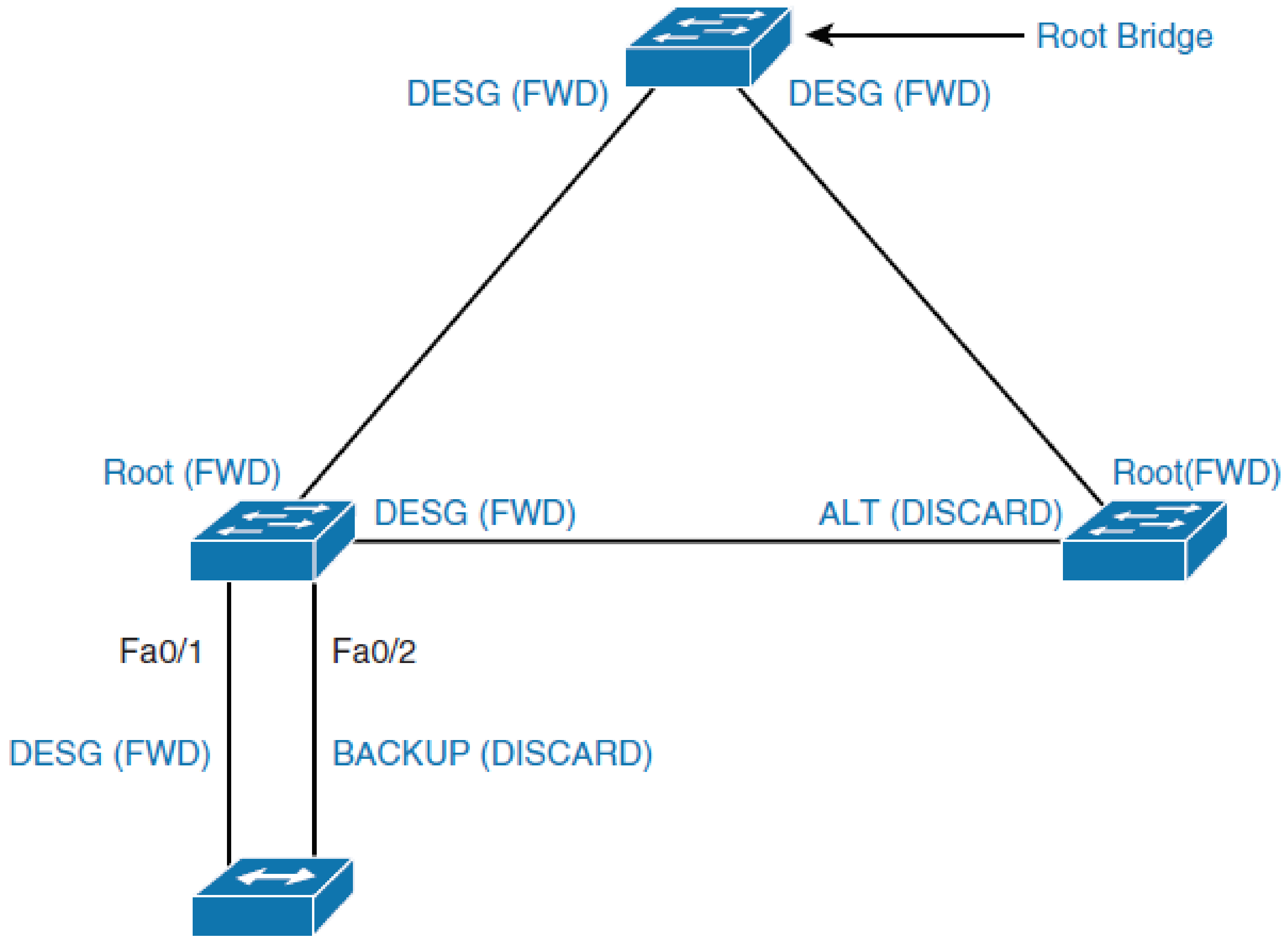
- **Disabled**
  - A disabled port has no role within the operation of spanning tree.

RSTP Port Roles

- **Backup (`Role = Back`)**
  - The backup port is an additional switch port on the designated switch with a redundant link to the shared segment for which the switch is designated.
  - The backup port has the discarding state (**BLK**) in active topology.

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----
Gi0/0               Desg FWD 4         128.1    Shr
Gi0/1               Desg FWD 200       128.2    Shr
Gi0/2               Back BLK 4         128.3    Shr
```

RSTP Port Roles



Automatic Detection of Topology Changes

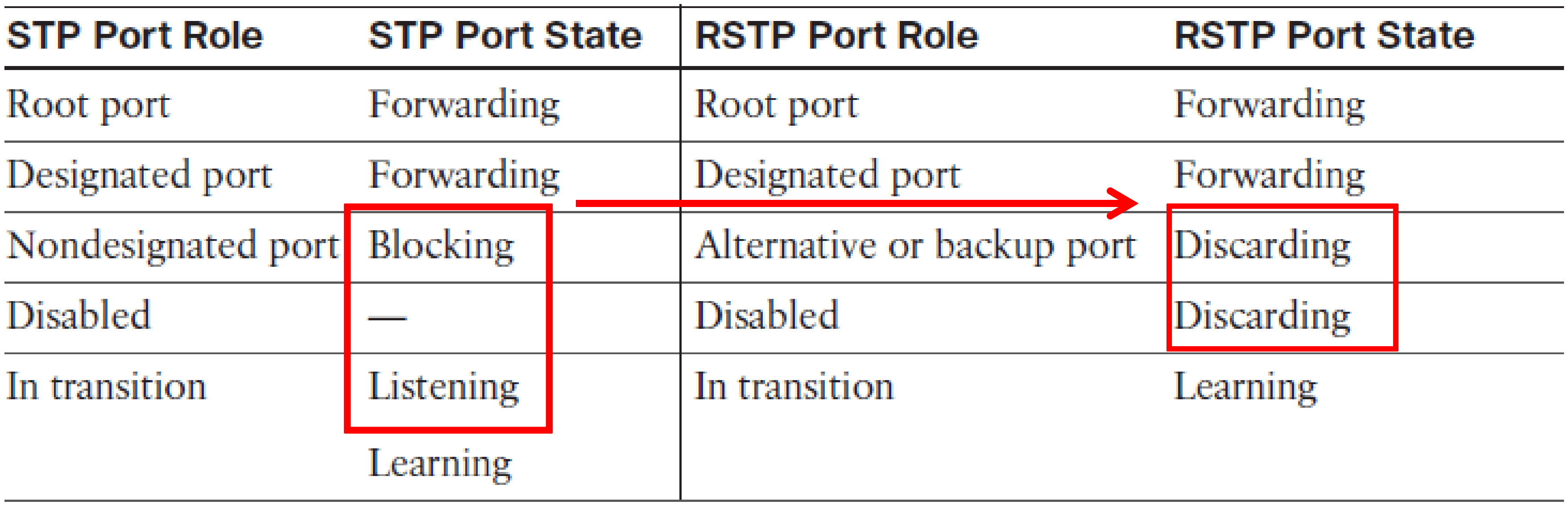
- In classical STP (802.1D), only the root bridge generates BPDUs (when all switches are synchronized), and these are forwarded downstream by other switches.
- In RSTP (802.1w), every switch generates BPDUs independently and sends them at Hello Time intervals (default: 2 seconds).
- This allows faster detection of topology changes and enables the **Proposal/Agreement mechanism** for rapid convergence.
- Thus, this feature improves network resilience and speeds up reconvergence when a failure occurs.

Proposal/Agreement Mechanism for Faster Convergence

- RSTP speeds up convergence using a **Proposal/Agreement handshake** between switches.
- When a switch detects a topology change, it sends a **Proposal BPDU** to its neighbour, suggesting a change in port state.
- The neighbouring switch responds with an **Agreement BPDU** if it confirms the change.
- This allows the switch to quickly update its port states without waiting for timers to expire, unlike the traditional STP method that relies on fixed timers.
- This handshaking mechanism dramatically reduces network recovery time.

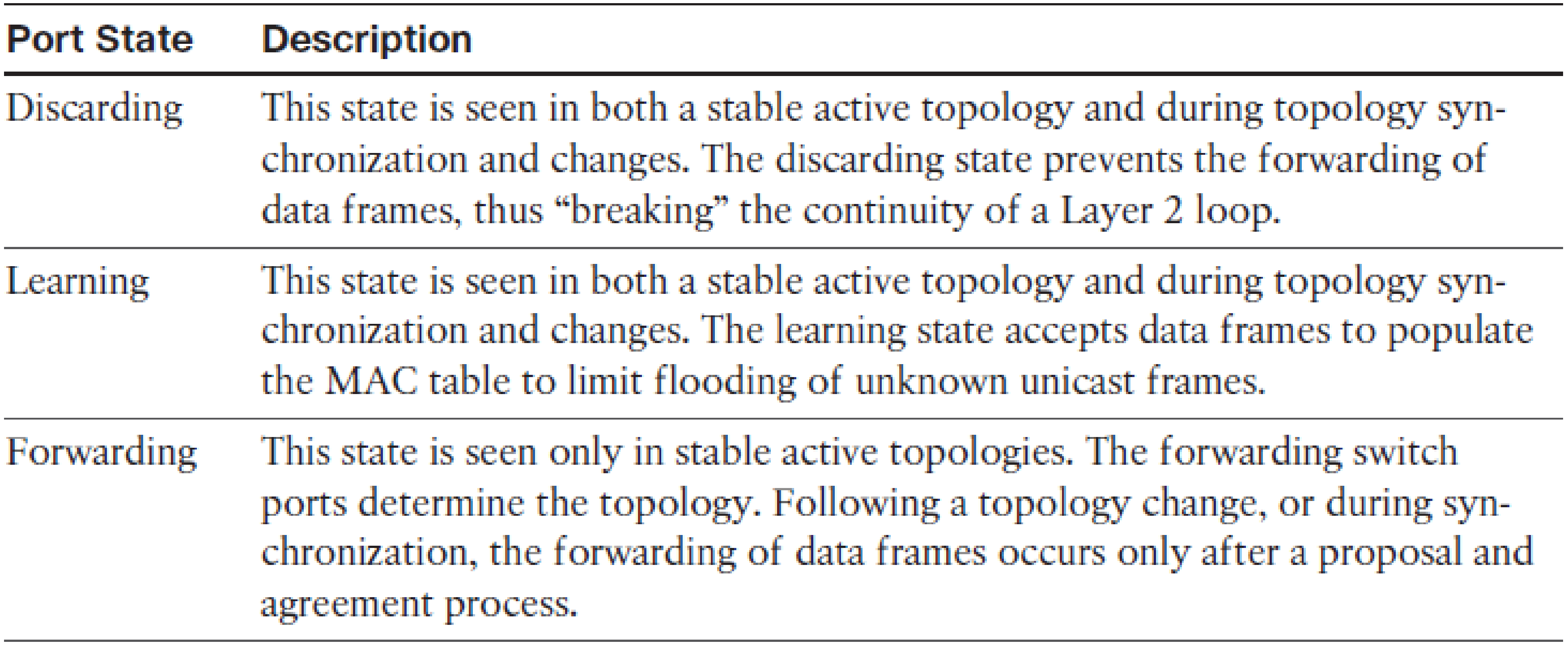
Comparison of RSTP and STP Port States

- in RSTP There is no listening state as there is with STP.
- **Blocking, Listening and Disabled** STP states are replaced with the **Discarding state**.
- In a stable topology, RSTP ensures that every root port and designated port transit to forwarding, while all alternate ports and backup ports are always in the discarding state.



STP Port Role	STP Port State	RSTP Port Role	RSTP Port State
Root port	Forwarding	Root port	Forwarding
Designated port	Forwarding	Designated port	Forwarding
Nondesignated port	Blocking	Alternative or backup port	Discarding
Disabled	—	Disabled	Discarding
In transition	Listening	In transition	Learning
	Learning		

RSTP Ports States



Port State	Description
Discarding	This state is seen in both a stable active topology and during topology synchronization and changes. The discarding state prevents the forwarding of data frames, thus "breaking" the continuity of a Layer 2 loop.
Learning	This state is seen in both a stable active topology and during topology synchronization and changes. The learning state accepts data frames to populate the MAC table to limit flooding of unknown unicast frames.
Forwarding	This state is seen only in stable active topologies. The forwarding switch ports determine the topology. Following a topology change, or during synchronization, the forwarding of data frames occurs only after a proposal and agreement process.

RSTP Topology Changes

- With RSTP, the TC propagation is now a one-step process. In fact, the initiator of the TC floods this information throughout the network, as opposed to 802.1D, where only the root did. RSTP no longer uses the specific TCN BPDUs unless a legacy bridge needs to be notified.
- This mechanism is much faster than the 802.1D equivalent.
- In just a few seconds, or a small multiple of hello times, most of the entries in the CAM tables of the entire network (VLAN) flush.
- When a switch receives a BPDU (with TC bit set) from a neighbor, it clears the MAC addresses learned on all its ports except the one that receives the topology change.

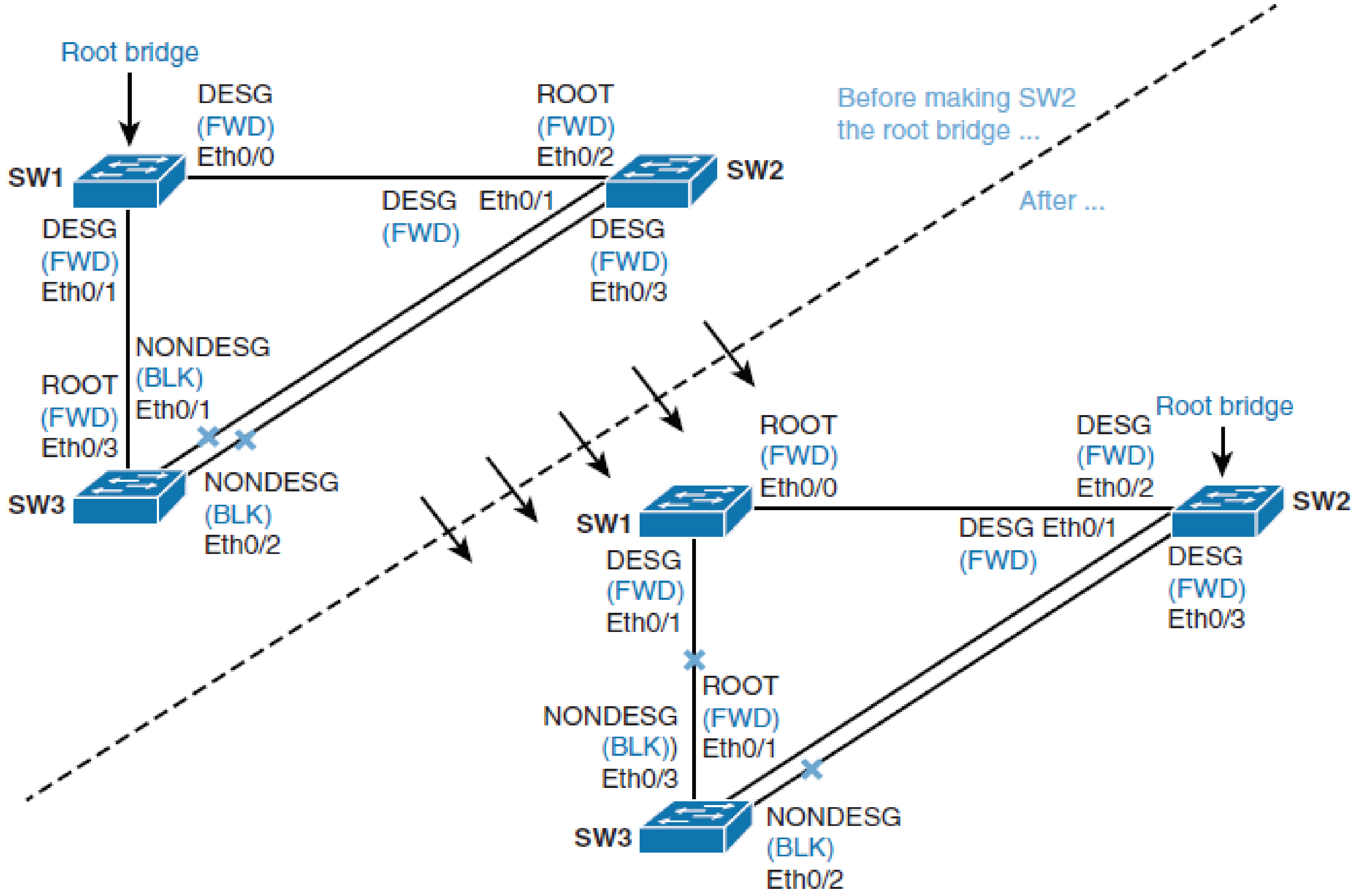
RSTP Topology Changes

- **Why does RSTP not consider link failure as a topology change?**
  - Loss of connectivity does not provide new paths in topology. If a switch loses the link to a **downstream switch**, the downstream switch either has an alternate path to the root bridge or it does not.
  - If the downstream switch has no alternate path, no action will be taken to improve convergence.
  - If the downstream switch has an alternate path, the downstream switch will unblock it and consequently generate its own BPDUs with the TC bit set.
  - Like with STP, PortFast-enabled ports do not create topology changes.

Configuring and Modifying STP Behavior

- Changing STP Priority
- The oldest switch will have the lowest MAC address because the lower MAC addresses were factory-assigned first.
- To manually set the root bridge, you can change a switch's priority
- It is highly recommended to configure the **distribution** or **core** switches to **become the root bridge**
- The priority can be a value between 0 and 65,535 in increments of 4,096. The default value is 32,768
- The better solution is to use `spanning-tree vlan` ***`vlan-id`*** **`root`** `{primary | secondary}` command.

Configuring and Modifying STP Behavior
(Changing STP Priority of SW2 to lowest)



Changing STP Priority - recommended

- It is not advised for the network to choose the root bridge by itself.
- If all switches have default STP priorities, the switch with the lowest MAC address will become the root bridge.
- The oldest switch will have the lowest MAC address because the lower MAC addresses were factory-assigned first.
- To manually set the root bridge, you can change a switch’s priority

Note:

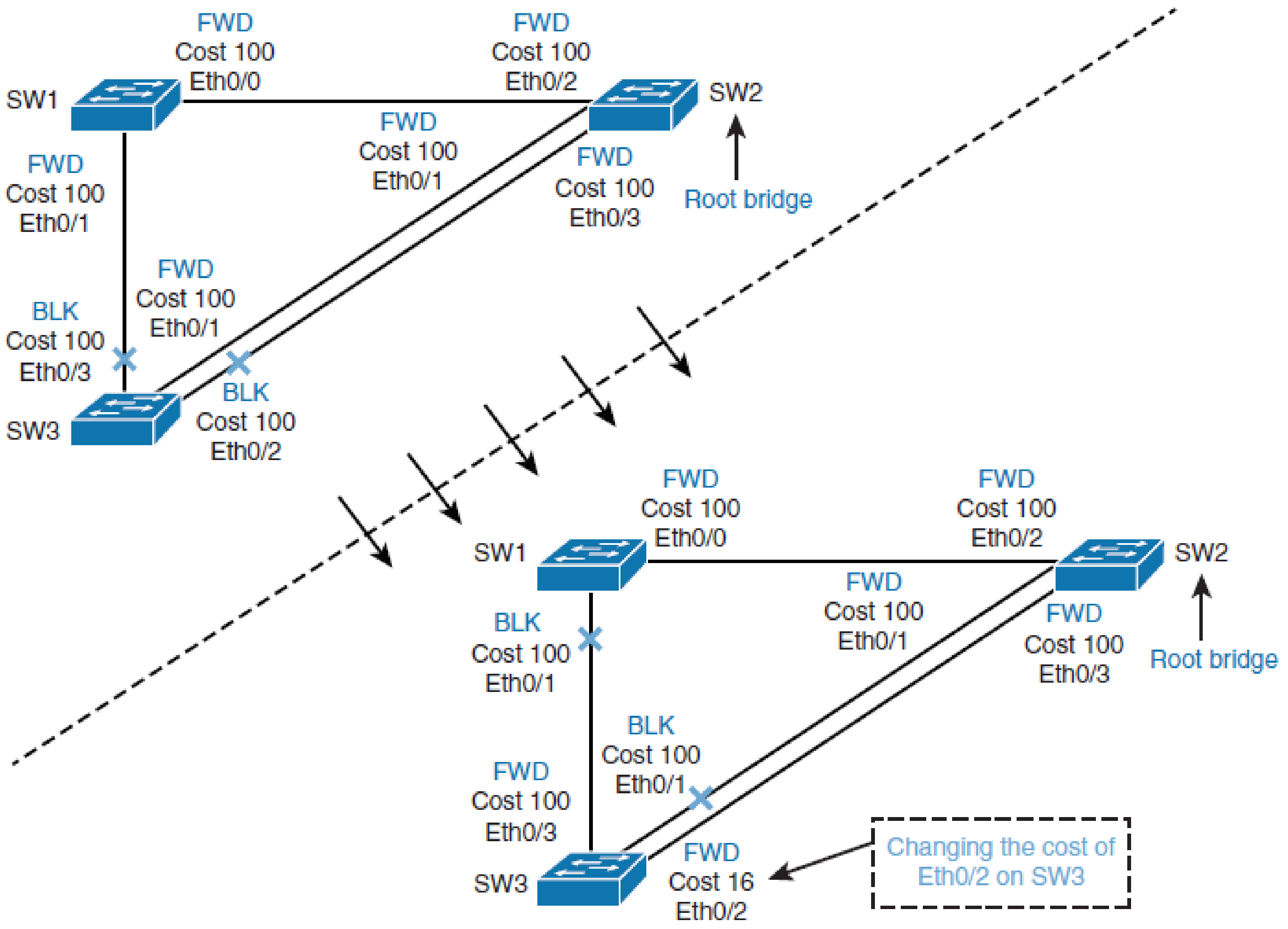
It is highly recommended to configure the distribution or core switches to become the root bridge.

Changing STP Priority

- The priority can be a value between 0 and 65,535, in increments of 4,096. The default value is 32,768.
- The better solution is to use **spanning-tree vlan** *vlan-id* **root** { **primary** | **secondary** } command.
- This command is actually a macro that lowers the switch’s priority number for it to become the root bridge.
- To configure the switch to become the root bridge for a specified VLAN, use the **primary** keyword.
- Use the **secondary** keyword to configure a secondary root bridge.
- The spanning-tree **root** command calculates the priority by learning the current root priority and lowering the 4096 value to it.

```
SW2(config)# spanning-tree vlan 1 root primary
```

STP Path Manipulation



STP Path Manipulation

- You can modify port cost by using the **`spanning-tree vlan`** *`vlan-list`* **`cost`** *`cost-value`* command.
- The cost value can be between 1 and 65,535.

```
DLS1(config)#
DLS1(config)#interface e0/1
DLS1(config-if)#spanning-tree cost ?
  <1-65535>  port path cost

DLS1(config-if)#spanning-tree cost 12
DLS1(config-if)#spanning-tree vlan 666 cost 333
```

STP Path Manipulation

```
DLS1(config-if)#do sh span vlan 110

VLAN0110
  Spanning tree enabled protocol rstp
  Root ID    Priority    32878
             Address     aabb.cc00.1100
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32878  (priority 32768 sys-id-ext 110)
             Address     aabb.cc00.1100
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------
Et0/0               Desg FWD 100       128.1    Shr
Et0/1               Desg FWD 12        128.2    Shr
Et0/2               Desg FWD 100       128.3    Shr
```

STP Path Manipulation

```
DLS1(config-if)#do sh span vlan 666

VLAN0666
  Spanning tree enabled protocol rstp
  Root ID    Priority    33434
             Address     aabb.cc00.1100
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33434  (priority 32768 sys-id-ext 666)
             Address     aabb.cc00.1100
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------
Et0/0               Desg FWD 100       128.1    Shr
Et0/1               Desg FWD 333       128.2    Shr
Et0/2               Desg FWD 100       128.3    Shr
```

STP Path Manipulation

- You can modify the port priority by using the **spanning-tree vlan** *vlan-list* **port-priority** *port-priority* command.
- The value of port priority can be between 0 and 255; the default is 128.
- A lower port priority means a more preferred path to the root bridge.

```
DLS1(config)#int e0/2
DLS1(config-if)#spanning-tree port-priority 192
DLS1(config-if)#spanning-tree vlan 120 port-priority 64
```

STP Path Manipulation

```
DLS1(config-if)#do sh span vlan 110

VLAN0110
  Spanning tree enabled protocol rstp
  Root ID    Priority    32878
             Address     aabb.cc00.1100
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32878  (priority 32768 sys-id-ext 110)
             Address     aabb.cc00.1100
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----
Et0/0               Desg FWD 100       128.1    Shr
Et0/1               Desg FWD 12        128.2    Shr
Et0/2               Desg FWD 100       192.3    Shr
```

STP Path Manipulation

```
DLS1(config-if)#do sh span vlan 120

VLAN0120
  Spanning tree enabled protocol rstp
  Root ID    Priority    32888
             Address     aabb.cc00.1100
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32888  (priority 32768 sys-id-ext 120)
             Address     aabb.cc00.1100
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----
Et0/0               Desg FWD 100       128.1    Shr
Et0/1               Desg FWD 12        128.2    Shr
Et0/2               Desg FWD 100        64.3    Shr
```

STP Timers

STP uses three different timers to ensure proper loop-free convergence. The three key STP timers and their default values are as follows:

```
ALS2# show span root
                                        Root    Hello Max Fwd
Vlan                 Root ID            Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0099         32867 5017.ff84.0a80         0    2   20  15
VLAN0100         32868 5017.ff84.0a80         0    2   20  15
VLAN0110         32878 5017.ff84.0a80         0    2   20  15
VLAN0120         32888 5017.ff84.0a80         0    2   20  15
VLAN0666         33434 5017.ff84.0a80         0    2   20  15
```

STP Timers

Hello Time

- The *Hello timer* determines the interval at which an STP-capable device sends configuration BPDUs.
- The device sends BPDUs at an interval of the Hello timer to check whether any link has failed.
- When the Hello timer is changed, the new value takes effect only after a new root bridge is elected.
- New root bridge includes the new Hello timer value in BPDUs it sends to non-root bridges.
- If the network topology changes, TCN BPDUs are immediately transmitted regardless of the Hello timer.

STP Timers

Forward Delay timer

- specifies the delay in a port state transition.
- When a link fails, STP recalculation is triggered and the spanning tree structure changes accordingly. However, new configuration BPDUs cannot be flooded immediately across the entire network. If the new root port and designated port forward data immediately, transient loops may occur. Therefore, STP defines a delay mechanism for port state transition. The newly selected root port and designated port must wait for two Forward Delay intervals before transitioning to the Forwarding state. During this period, the new configuration BPDUs can be transmitted over the network, preventing transient loops.

STP Timers

Forward Delay timer

- The Forward Delay timer defines the time that is spent in Listening or Learning state.
- The default Forward Delay value is 15 seconds.
- This means that the port stays in Listening state for 15 seconds and then stays in Learning state for another 15 seconds before transitioning to the Forwarding state.
- The port in Listening or Learning state does not forward user traffic, effectively preventing transient loops.

STP Timers

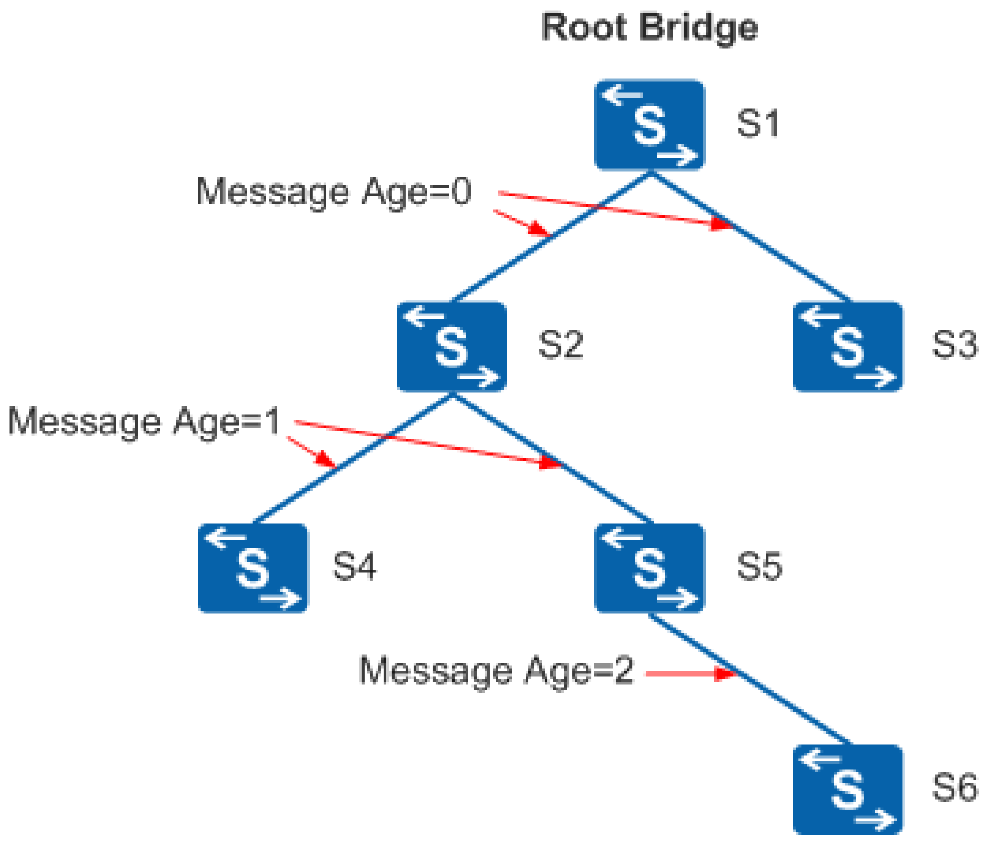
Max Age timer

- The Max Age timer specifies the aging time of BPDUs. This parameter is configurable on the root bridge.
- The Max Age value is encapsulated in configuration BPDUs and transmitted on the entire network to ensure consistency. Upon receipt of a configuration BPDU, a non-root bridge compares the **Message Age value** with the **Max Age value** in the received configuration BPDU.
  - If the Message Age value is smaller than or equal to the Max Age value, the non-root bridge forwards the configuration BPDU.
  - If the Message Age value is greater than the Max Age value, the non-root bridge discards the configuration BPDU. In this case, the network is considered too large and the non-root bridge disconnects from the root bridge.

STP Timers

Max Age timer

- If the configuration BPDU is sent from the root bridge, the Message Age value is 0.
- Otherwise, the Message Age value is the total time required to transmit the BPDU from the root bridge to the local bridge, including the transmission delay.
- The Message Age value of a configuration BPDU is incremented by 1 second each time the configuration BPDU passes through a bridge:



The Message Age **is measured in seconds**, not just as a hop count

STP Timers

- The transition between port states takes from 30 to 50 seconds, depending on the topology change.
- This can be adjusted with STP timers:
  - **Hello Time** can be tuned between 1 and 10 seconds,
  - **Forward Delay** between 4 and 30 seconds,
  - and **Maximum Age** between 6 and 40 seconds.
- To manually configure timers, use the command:

spanning-tree [**vlan** *vlan-id*] { **hello-time** | **forward-time** | **max-age** } *seconds*

STP Timers optimalization

- To prevent frequent network flapping, ensure that the values of the Hello timer, Forward Delay timer, and Max Age timer of the root switch conform to the following formulas:
- **2 x (Forward Delay timer - 1.0 second) >= Max Age timer**
- **Max Age timer >= 2 x (Hello timer + 1.0 second)**
- You are advised to set the three parameters on devices consistently on an STP network.
- Generally, you are not advised to directly change these timers.

Implementing STP Stability Mechanisms

Cisco Spanning Tree Protocol Toolkit

Provides tools to better manage STP.

The key features of are as follows:

- **UplinkFast:** Enables fast uplink failover on access switch
- **BackboneFast:** Enables fast convergence in distribution or core layer when STP change occurs
- **PortFast:** Configures access port to transition directly to forwarding state

Cisco Spanning Tree Protocol Toolkit

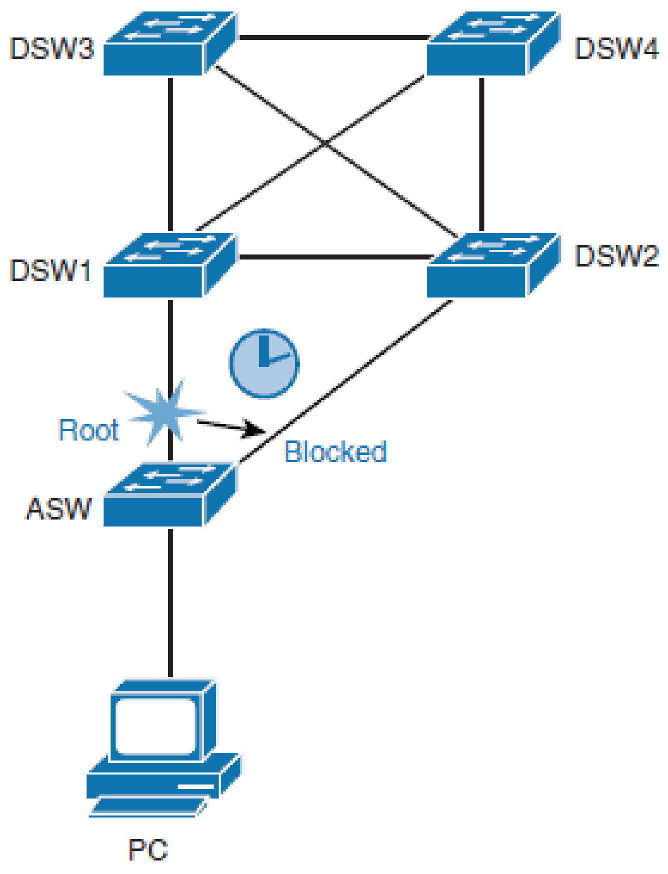
The key features of the Cisco STP Toolkit that ensure STP stability are as follows:

- **BPDU Guard**
  - Disables the PortFast-enabled port if a BPDU is received
- **BPDU Filter**
  - Suppresses BPDUs on switch ports
- **Root Guard**
  - Prevents external switches to become a root bridge
- **Loop Guard**
  - Prevents an alternate port from becoming the designated port if no BPDUs are received

Use UplinkFast

- If forwarding uplink fails, it will take 30 to 50 seconds for the other uplink to take over.
- UplinkFast is a Cisco proprietary solution that greatly reduces convergence time.
- **The UplinkFast feature is based on the definition of an uplink group. On a given switch, the uplink group consists of the root port and all the ports that provide an alternate connection to the root bridge. If the root port fails, which means if the primary uplink fails, a port with the next lowest cost from the uplink group is selected to immediately replace it.**
- The total time to recover the primary link failure will normally be less than 1 second.

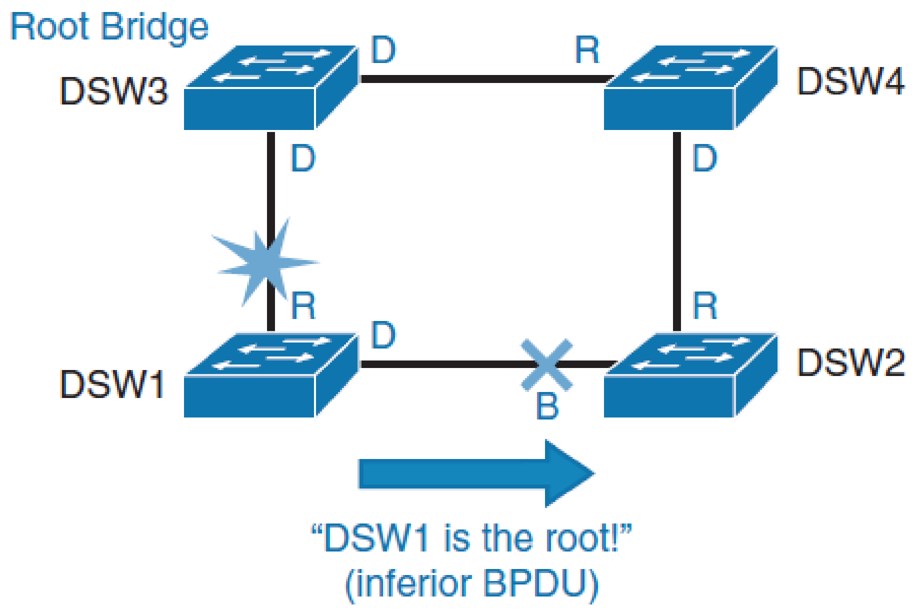
Use UplinkFast



- UplinkFast is a Cisco proprietary feature
- By default, UplinkFast is disabled.
- To enable UplinkFast, use the following command:
  - ASW(config)# **spanning-tree uplinkfast**
- With RSTP, the UplinkFast mechanism is already integrated into the protocol in a standards-based way.

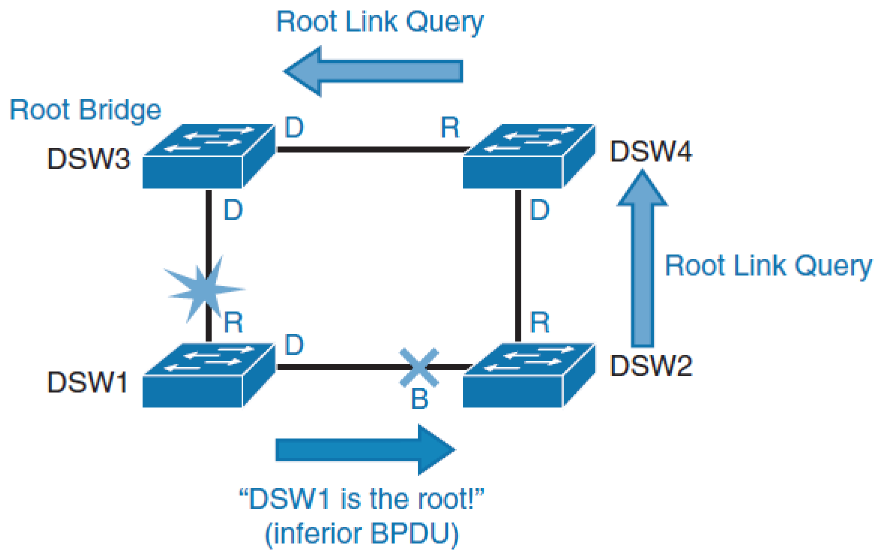
Use BackboneFast

- When an **indirect link failure occurs**, BackboneFast checks whether an alternative path exists to the root bridge.
- Indirect failure is when a link that is not directly connected to a switch fails.



Use BackboneFast

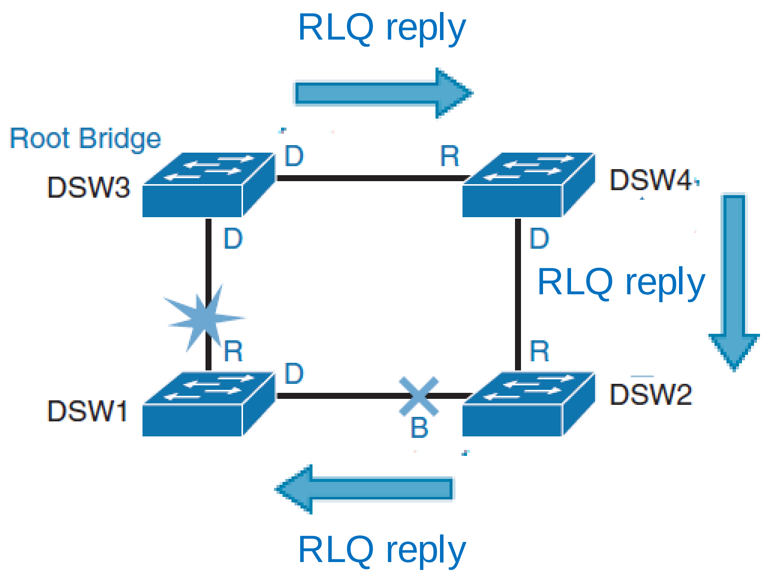
- As shown in Figure, DSW3 is the **root bridge**, and DSW2 is the one blocking DSW1’s alternate path to DSW3.
- When DSW1’s root port fails, DSW1 declares itself the root bridge and starts sending BPDUs to all switches it is connected to (in this case, only DSW2).



- These BPDUs are **inferior**. When a switch receives an inferior BPDU on a blocked port and **BackboneFast enabled**, it runs a procedure to validate that it still has an active path to the currently known root bridge:
  - After the switch identifies potential alternative ports, it starts sending RLQs (Root Link Queries).
  - By sending these queries, it finds out whether upstream switches have a path to the root bridge

Use BackboneFast

- When a switch, which is either the root bridge or has a connection to the root bridge, receives an RLQ, the switch sends back an RLQ reply.
- Otherwise, an RLQ gets forwarded until it gets to a switch that is the root bridge or has a connection to the root bridge.



- If exchange of RLQ messages results in validation that the root bridge (DSW3) is still accessible, the switch (DSW2) starts sending existing root bridge information to the bridge that lost connectivity through its root port (DSW1).
- If this validation fails, DSW2 can start the root bridge election process. In either of these cases, if validation is successful or not, maximum age time is shortened.

Use BackboneFast

Normally a switch must wait for the maximum age timer to expire before responding to the inferior BPDUs.

However, BackboneFast searches for an alternative path:

- If the inferior BPDU arrives on a port that is **blocked**, the switch assumes that the root port and all other blocked ports are an alternative path.
- If the inferior BPDU arrives on a port that is root, the switch assumes all blocked are an alternate path.
- If no ports are blocked, the switch assumes that it lost connectivity with the root bridge and considers itself as the root bridge.

After the switch identifies potential alternative ports, it starts sending RLQs (request link queries). By sending these queries, it finds out whether upstream switches have a path to the root bridge.

Use BackboneFast

- To configure BackboneFast, use the following command:

```
DSW1(config)# spanning-tree backbonefast
```

- By default, BackboneFast is disabled.
- To verify the current BackboneFast state, issue the following command:

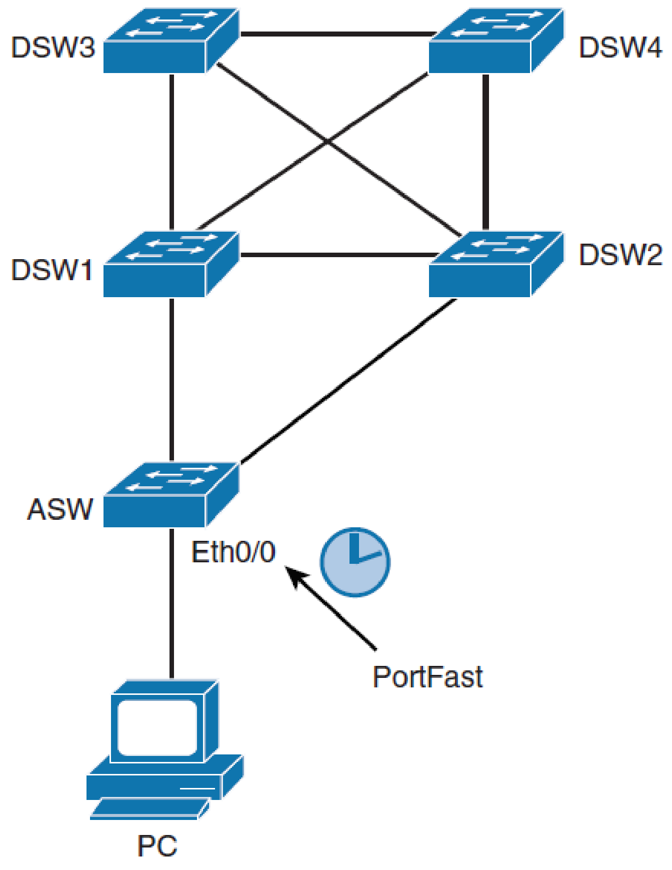
```
DSW1# show spanning-tree backbonefast
BackboneFast is enabled
```

- BackboneFast was implemented also into RSTP.
- RSTP implementation differs a bit from BackboneFast. Whereas BackboneFast relies on RLQ messages to validate the current root bridge, RSTP relies on cached information.

Use PortFast

- When PortFast is enabled, the port transitions immediately from blocking to forwarding.
- PortFast should be enabled on **access layer switches** where the hosts are connected.
- An additional benefit of using PortFast is that **TCN BPDU**s are not sent when a switch port in PortFast mode goes up or down.
- By default, PortFast is disabled on all switch ports.
- You can configure PortFast in two ways: per port and globally.
  - If you configure PortFast globally, all ports that are configured as access ports automatically become PortFast enabled, and the port will immediately transition to forwarding.
    - If a port does receive a BPDU, that port will go into **blocking mode**.
  - If you configure PortFast per port
    - The port will be PortFast enabled even if it receives BPDUs

Use PortFast



```
ASW(config-if)# spanning-tree portfast
! Enables PortFast on per-port basis
ASW(config)# spanning-tree portfast default
! Enables PortFast on ALL switch ports that are defined as access
ASW# show spanning-tree interface ethernet 0/0 portfast
! Displays current state of PortFast
```

PortFast Configuration for a Trunk

- Never use the PortFast feature on switch ports that connect to other switches, hubs, or routers.
- You can also enable PortFast on trunk ports.
- This is useful if you have a trunk enabled for a host such as a server that needs multiple VLANs.
- To enable a port for PortFast on an interface that connects to such a server, use the following interface configuration commands:

```
ASW(config-if)# spanning-tree portfast trunk
! To display the current status of PortFast, use the following command:
ASW# show spanning-tree interface ethernet 0/0 portfast
VLAN0001            enabled
```

Securing PortFast Interface with BPDU Guard

- BPDU Guard protects the integrity of ports that are PortFast enabled.
- If any BPDU is received on a PortFast-enabled port, that port is put into **err-disabled state**.
- That means the port is shut down and must be manually re-enabled or automatically recovered through the error-disabled timeout function.

```
%SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port Et0/0 with BPDU Guard enabled.
Disabling port.
%PM-4-ERR_DISABLE: bpduguard error detected on Et0/0, putting Et0/0 in err-disable
state
```

- It is highly recommended to always enable BPDU Guard on all PortFast-enabled ports.

Securing PortFast Interface with BPDU Guard

- By default, BPDU Guard is disabled on all switch ports.
- BPDU Guard can be configured in two ways, globally and per port.

```
ASW(config-if)# spanning-tree bpduguard enable
! Configures BPDU Guard on a port
ASW(config)# spanning-tree portfast bpduguard default
! Configures BPDU Guard on all switch ports that have PortFast enabled
ASW# show spanning-tree summary totals
! Verifies BPDU Guard configuration
```

- Global configuration is conditional: If the port is not PortFast enabled, BPDU Guard will not be activated.

Disabling STP with BPDU Filter

- BPDUs are sent on all ports, even if they are PortFast enabled.
- You should always run STP to prevent loops.
- However, in special cases, you need to prevent BPDUs from being sent out. You can achieve that by using BPDU Filter.
- Configuring BPDU Filter so that all configuration BPDUs received on a port are dropped can be useful for service provider environments, where a service provider provides Layer 2 Ethernet access for customers. Ideally, the service provider does not want to share any spanning-tree information with customers, because such sharing might jeopardize the stability of the service provider's internal spanning-tree topology.
- By configuring **PortFast** and **BPDU Filter** on each customer access port, the service provider will not send any configuration BPDUs to customers and will ignore any configuration BPDUs sent from customers!

Disabling STP with BPDU Filter

```
Switch(config-if)#spanning-tree bpdufilter enable
! Enables BPDU Filter on a specific switch port

Switch(config)#spanning-tree portfast bpdufilter default
! Enables BPDU Filter on all switch ports that have
PortFast enabled

Switch# show spanning-tree totals
! Verify global BPDU Filter configuration

Switch# show spanning-tree interface e0/0 detail
! Verify BPDU Filter configuration on a specific port
```

Disabling STP with BPDU Filter

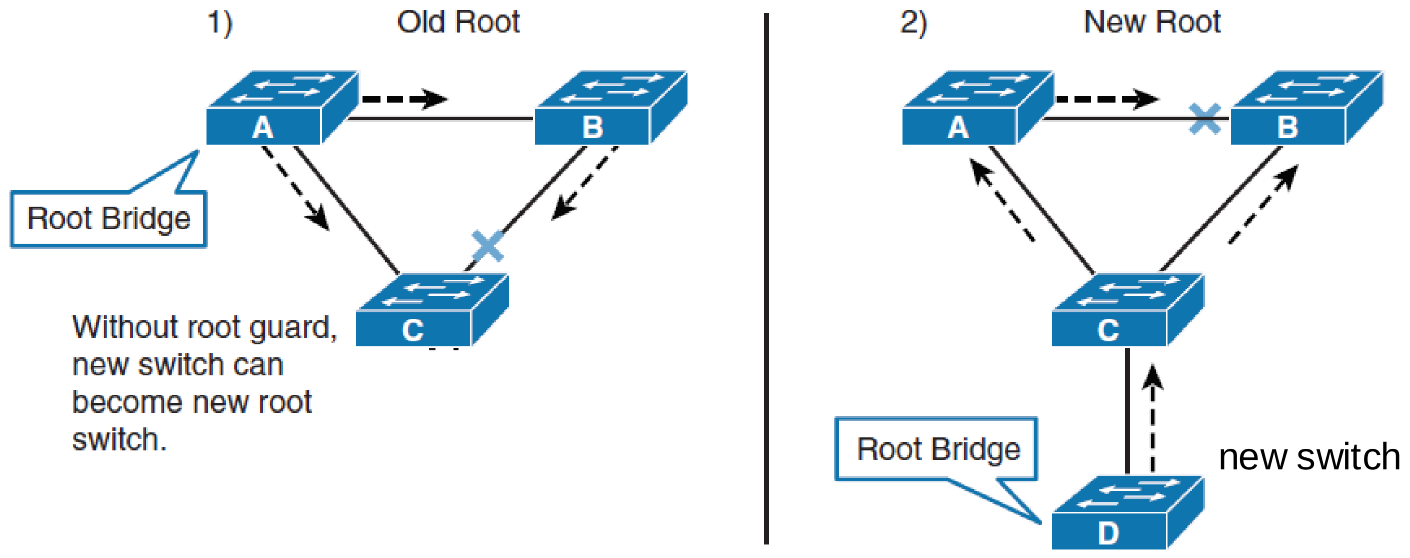
BPDU Filter behaves differently if applied globally or on a per-port basis:

- When enabled **globally**, BPDU Filter has these attributes:
  - It affects all operational PortFast ports on switches that do not have BPDU Filter configured on the individual ports.
  - If BPDUs are detected, the port loses its PortFast status, BPDU Filter is disabled, and the STP sends and receives BPDUs on the port as it would with any other STP port on the switch.
  - Upon startup, the port transmits ten BPDUs. If this port receives any BPDUs during that time, PortFast and BPDU Filter are disabled.
- When enabled on an **individual port**, BPDU Filter has these attributes:
  - It ignores all BPDUs received.
  - It sends no BPDUs.

Use Root Guard

- The Root Guard feature forces an interface to become a designated port to prevent surrounding switches from becoming a root switch.
- In other words, Root Guard provides a way to enforce the root bridge placement in the network.
- In other words, Root Guard prevents a Designated Port from becoming a Root Port.
- If a port on which the Root Guard feature receives a superior BPDU, it moves the port into a root-inconsistent state (effectively equal to a listening state), thus maintaining the current Root Bridge status.

Use Root Guard



- **Best Practice**: Apply Root Guard on links connecting to downstream switches but not on uplinks toward the root bridge
- **Final Thought**: Root Guard is essential in ensuring the intended Root Bridge remains stable and preventing rogue devices or misconfigurations from disrupting the STP topology

Configuring and Verifying Root Guard

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet 5/8
Switch(config-if)# spanning-tree guard root
Switch(config-if)# end
Switch# show running-config interface FastEthernet 5/8
Building configuration... Current configuration: 67 bytes ! interface
  FastEthernet5/8 switchport mode access spanning-tree guard root end !
```

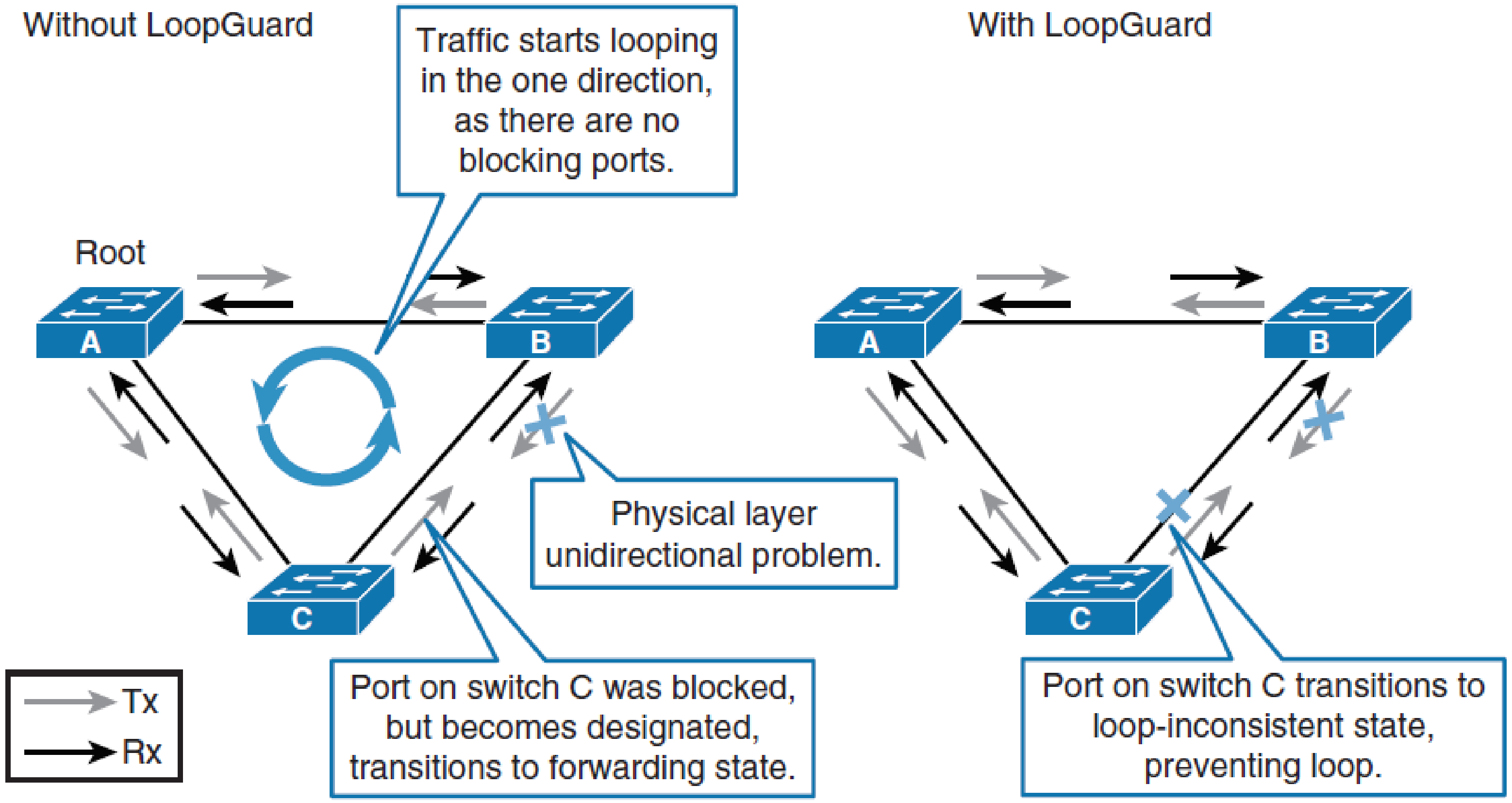
```
%SPANTREE-2-ROOTGUARDBLOCK: Port 1/1 tried to become non-designated in VLAN 77.
Moved to root-inconsistent state
```

```
Switch# show spanning-tree inconsistentports
Name Interface Inconsistency
VLAN0001     FastEthernet3/1     Port Type Inconsistent
VLAN0001     FastEthernet3/2     Port Type Inconsistent
VLAN1002     FastEthernet3/1     Port Type Inconsistent
VLAN1002     FastEthernet3/2     Port Type Inconsistent
```

Loop Guard Overview

- STP relies on continuous reception or transmission of BPDUs based on the port role.
- The designated port transmits BPDUs, and the non-designated port just receives BPDUs.
- When one of the ports in a physically redundant topology no longer receives BPDUs, the STP conceives that the topology is loop free.
- Eventually, a port in blocking state - from the alternate or backup role becomes designated role and moves to a forwarding state.
- This situation can create a loop.
- The Loop Guard feature makes additional checks.
- If BPDUs are not received on a non-designated port, and Loop Guard is enabled, that port is moved into the STP loop inconsistent blocking state, instead of the **listening/learning/forwarding** state.

Loop Guard Overview


Without LoopGuard
Traffic starts looping in the one direction, as there are no blocking ports.
With LoopGuard
Root
A
B
C
A
B
C
Physical layer unidirectional problem.
Tx
Rx
Port on switch C was blocked, but becomes designated, transitions to forwarding state.
Port on switch C transitions to loop-inconsistent state, preventing loop.

Loop Guard Overview

- When the Loop Guard blocks an inconsistent port, this message is logged:

```
%SPANTREE-2-LOOPGUARD_BLOCK: LoopGuard blocking port FastEthernet0/24 o VLAN0050.
```

- Once the BPDUs are received again on a port in a loop-inconsistent STP state, the port transitions into another STP state (listening/learning/forwarding).
- After recovery, this message is logged:

```
%SPANTREE-2-LOOPGUARD_UNBLOCK: LoopGuard unblocking port FastEthernet0/24
VLAN0050.
```

Loop Guard Placement

- The Loop Guard feature is enabled on a per-port basis.
- However, as long as Loop Guard blocks the port on the STP level, Loop Guard blocks inconsistent ports on a per-VLAN basis.
- By default, Loop Guard is disabled.
- You can configure Loop Guard globally or on a per-port basis.

Loop Guard Placement

- If you enable Loop Guard globally, then effectively, it is enabled on all point-to-point (P2P) links.

```
Switch(config)# interface Ethernet 0/0
Switch(config-if)# spanning-tree guard loop
! Enables Loop Guard on a per-interface basis
Switch(config)# spanning-tree loopguard default
! Enables Loop Guard globally on all point-to-point links
```

STP Link Type	Duplex Mode	Used for	BPDU Behavior	Loop Guard?
Point-to-Point (P2P)	Full-Duplex	Switch-to-Switch	Direct BPDU exchange	✅ Yes
Shared	Half-Duplex	Hubs, Legacy Networks	Shared BPDU communication	❌ No
Edge (PortFast)	N/A	PCs, Servers	No BPDUs expected	❌ No

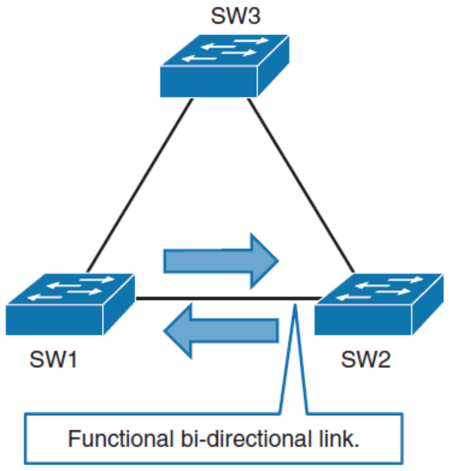
Loop Guard vs Root Guard

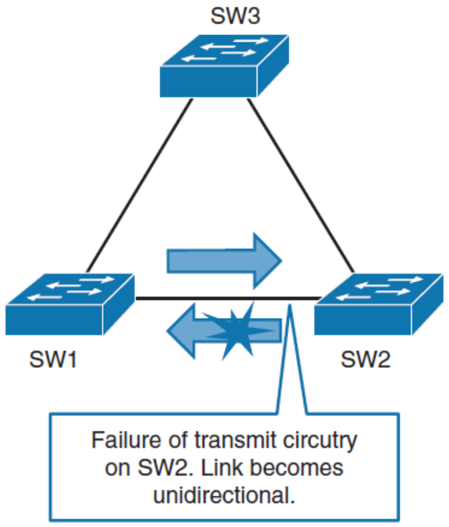
- The Root Guard is mutually exclusive with the Loop Guard.
- The Root Guard is used on designated ports, and it does not allow the port to become non-designated.
- The Loop Guard works on non-designated ports and does not allow the port to become designated through the expiration of maximum age.
- The Root Guard cannot be enabled on the same port as the Loop Guard.
- When the Loop Guard is configured on the port, it disables the Root Guard configured on the same port.

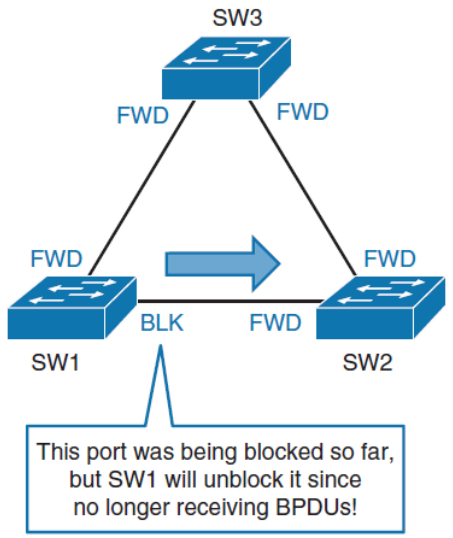
Use UDLD

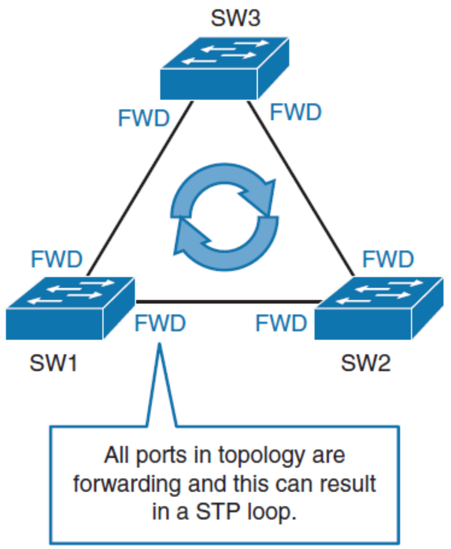
- Unidirectional links can cause spanning-tree topology loops.
- Unidirectional Link Detection (UDLD) enables devices to detect when a unidirectional link exists and also to shut down the affected interface.
- UDLD is useful on a fiber port to prevent network issues resulting in miswiring at the patch panel causing the link to be in up/up status but the BPDUs are lost.

UDLD Overview

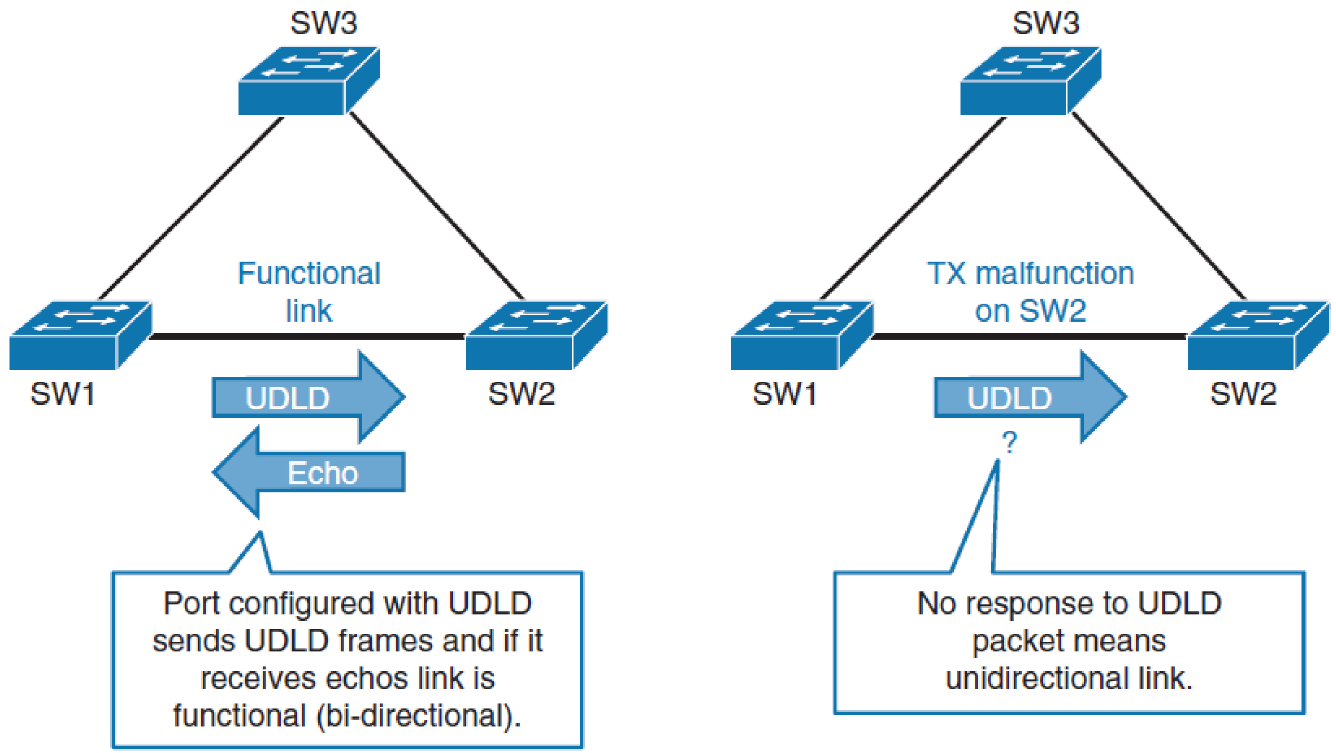



SW3
SW1
SW2
Failure of transmit circutry on SW2. Link becomes unidirectional.


SW3
FWD
FWD
FWD
FWD
BLK
FWD
SW1
SW2
This port was being blocked so far, but SW1 will unblock it since no longer receiving BPDUs!


SW3
FWD
FWD
FWD
FWD
FWD
FWD
SW1
SW2
All ports in topology are forwarding and this can result in a STP loop.

UDLD Overview


SW3
Functional link
SW1
SW2
UDLD
Echo
Port configured with UDLD sends UDLD frames and if it receives echos link is functional (bi-directional).
SW3
TX malfunction on SW2
SW1
SW2
UDLD
?
No response to UDLD packet means unidirectional link.

UDLD Overview

- UDLD is a Layer 2 protocol that works with the Layer 1 mechanisms to determine the physical status of a link.
- Both UDLD peers discover each other by exchanging special frames that are sent to well-known MAC address 01:00:0C:CC:CC:CC
- In an EtherChannel bundle, UDLD will error-disable only the physical link that has failed.
- UDLD messages are sent at regular intervals. This timer can be modified. The default setting varies between platforms. The typical value is 15 seconds.
- UDLD is a Cisco proprietary protocol that is also defined in RFC 5171.

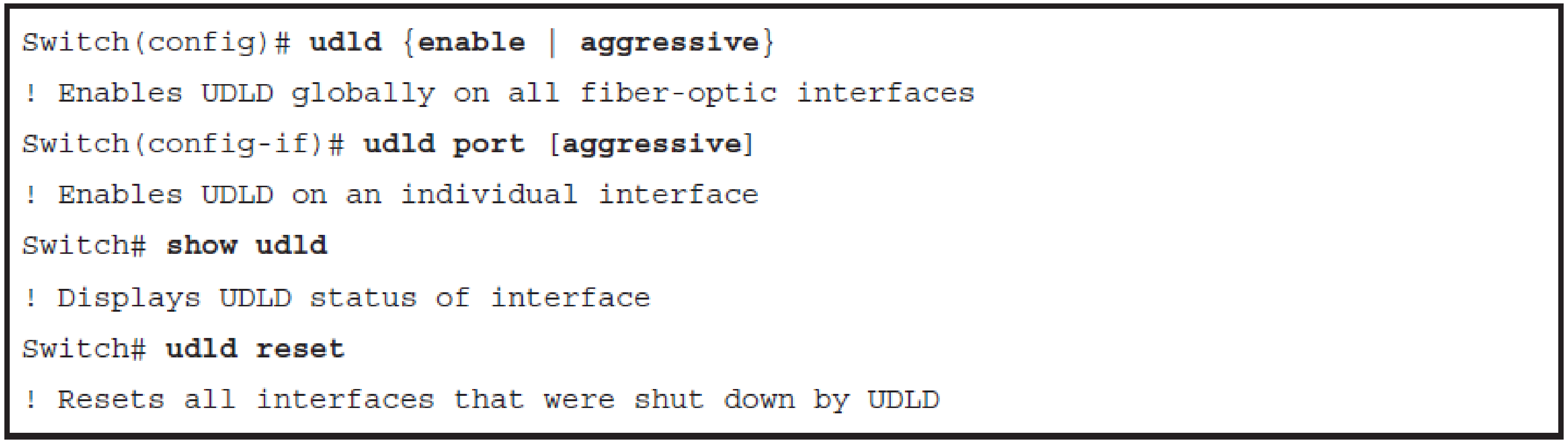
UDLD Operation

After UDLD detects an unidirectional link, it can take two courses of action, depending on configured mode:

- **Normal mode**
  - When an unidirectional link is detected, the port is allowed to continue its operation.
  - UDLD just marks the port as having an undetermined state. A syslog message is generated.
- **Aggressive mode**
  - When an unidirectional link is detected, the switch tries to re-establish the link.
  - It sends one message a second, for 8 seconds. If none of these messages is sent back, the port is placed in error-disabled state.

UDLD Configuration

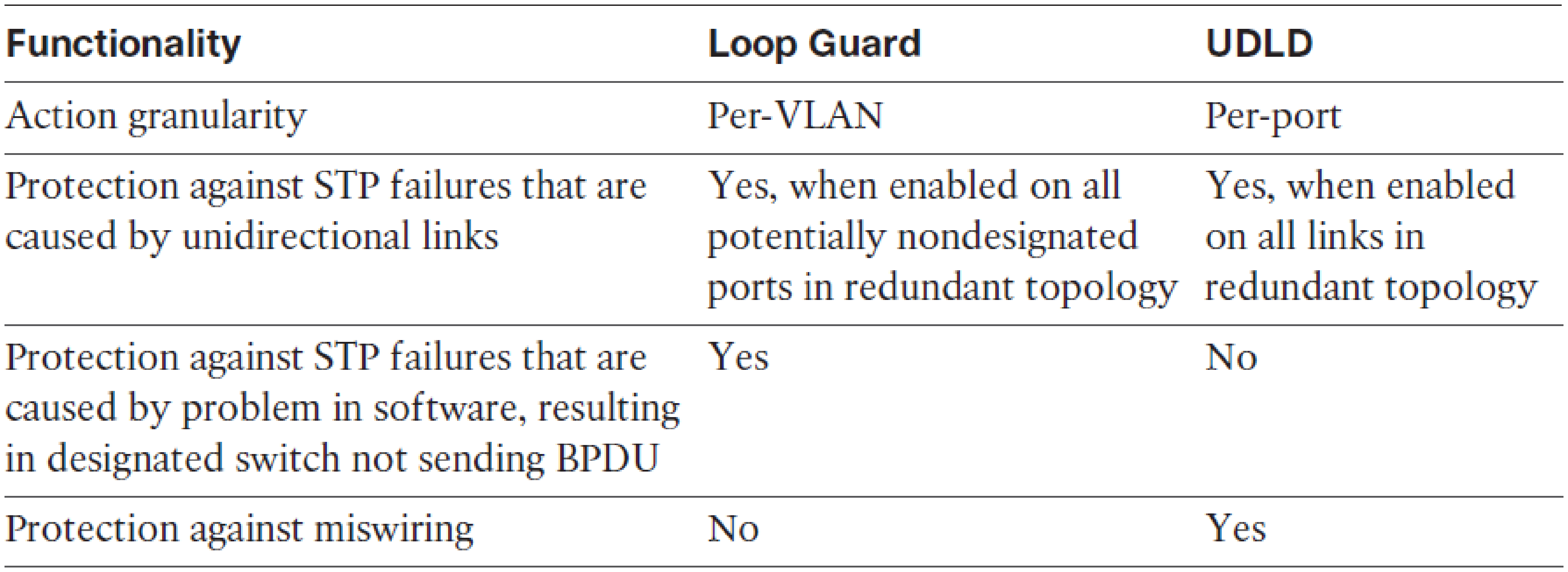
- As with other commands, like PortFast, you can enable UDLD on a per-port basis or globally.
- It is supported only at the fiber ports.



```
Switch(config)# udld {enable | aggressive}
! Enables UDLD globally on all fiber-optic interfaces
Switch(config-if)# udld port [aggressive]
! Enables UDLD on an individual interface
Switch# show udld
! Displays UDLD status of interface
Switch# udld reset
! Resets all interfaces that were shut down by UDLD
```

- Use the **udld reset** command to reset all the interfaces that were shut down by UDLD.

Comparing Loop Guard with UDLD



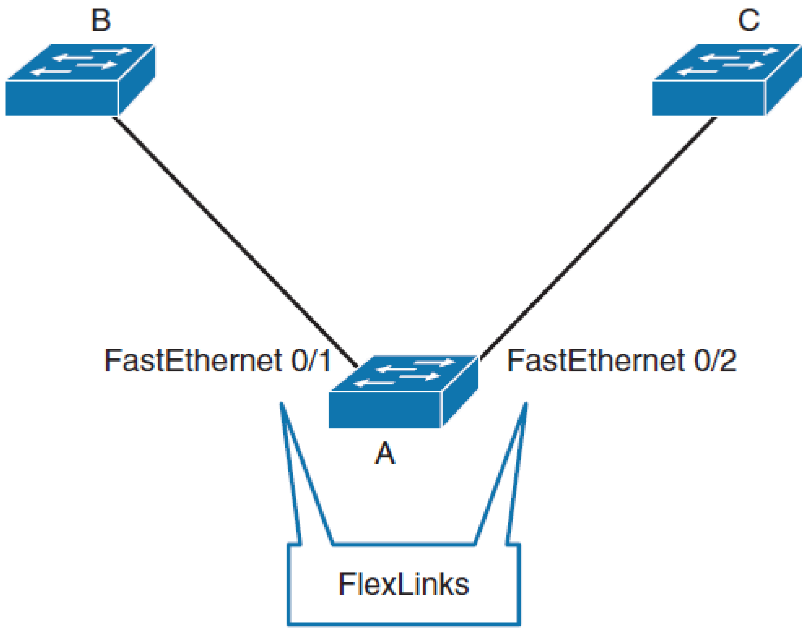
Functionality	Loop Guard	UDLD
Action granularity	Per-VLAN	Per-port
Protection against STP failures that are caused by unidirectional links	Yes, when enabled on all potentially nondesignated ports in redundant topology	Yes, when enabled on all links in redundant topology
Protection against STP failures that are caused by problem in software, resulting in designated switch not sending BPDU	Yes	No
Protection against miswiring	No	Yes

UDLD Recommended Practices

- Typically, it is deployed on any fiber-optic interconnection.
- Use UDLD aggressive mode for best protection.
- Turn on in global configuration to avoid operational errors and misses.

Use FlexLinks (alternative to STP)

- **FlexLinks** are a pair of a Layer 2 interfaces, where one interface is configured to act as a backup to the other.
- FlexLinks provide link-level redundancy that is an alternative to STP.
- STP is automatically disabled on FlexLinks interfaces.



- STP is turned off - one link is active, the other one on standby.
- Failover can be as low as 50ms.
- In example switches A and B are not aware of FlexLinks.

FlexLinks Configuration and Verification

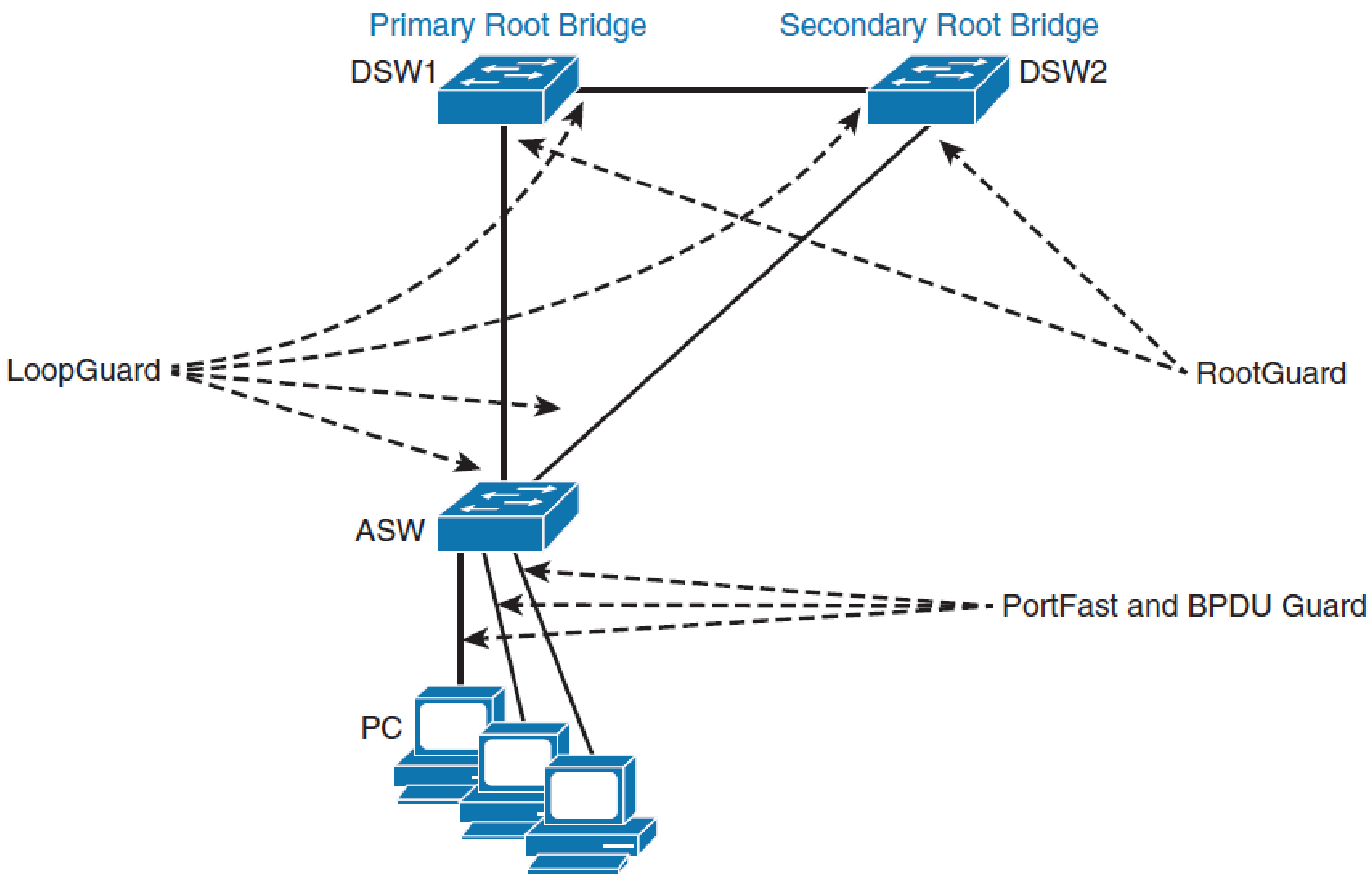
```
Switch(config)# interface FastEthernet0/1
Switch(config-if)# switchport backup interface FastEthernet0/2
! Configures an interface with a backup interface

Switch# show interface switchport backup
Switch Backup Interface Pairs:
Active Interface Backup Interface State
------------------------------------------------------
FastEthernet0/1 FastEthernet0/2 Active Up/Backup Standby
! Displays all FlexLinks configured on the switch and the state of each active and
  backup interface (up or standby mode)
```

FlexLinks Guidelines

- You can configure only one FlexLinks backup link for any active link
- An interface can belong to only one FlexLinks pair.
- Neither of the links can be a port that belongs to an EtherChannel. However, you can configure two port channels as FlexLinks.
- A backup link does not have to be the same type (Fast Ethernet, Gigabit Ethernet, or port channel) as the active link.
- STP is disabled on FlexLinks ports

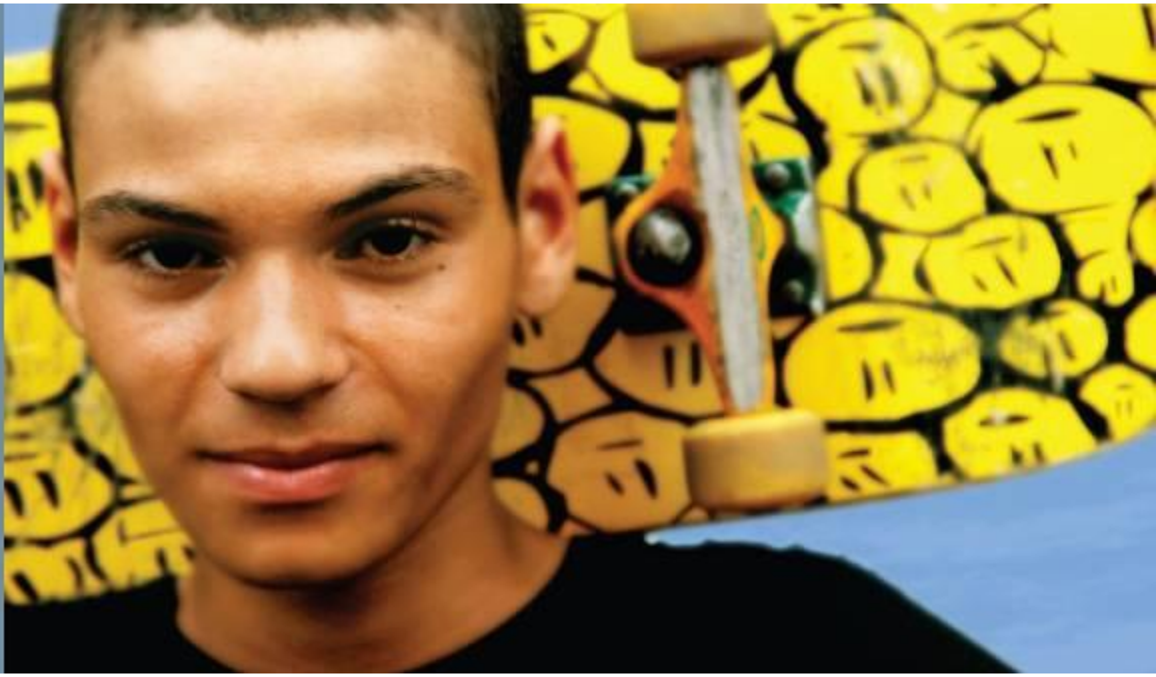
STP Stability Mechanisms Recommendations



STP Stability Mechanisms Recommendations

- **PortFast:** Apply to all end-user ports. To secure PortFast-enabled ports, always combine PortFast with **BPDU Guard**.
- **Root Guard:** Apply to all ports where root is never expected.
- **Loop Guard:** Apply to all ports that are or can become non-designated.
- **UDLD:** The UDLD protocol enables devices to monitor the physical configuration of the cables and detect when a unidirectional link exists.
- Depending on the security requirements of an organization, the port security feature can be used to restrict the ingress traffic of a port by limiting the MAC addresses that are allowed to send traffic into the port.

Configuring Multiple Spanning Tree Protocol (**MST**)

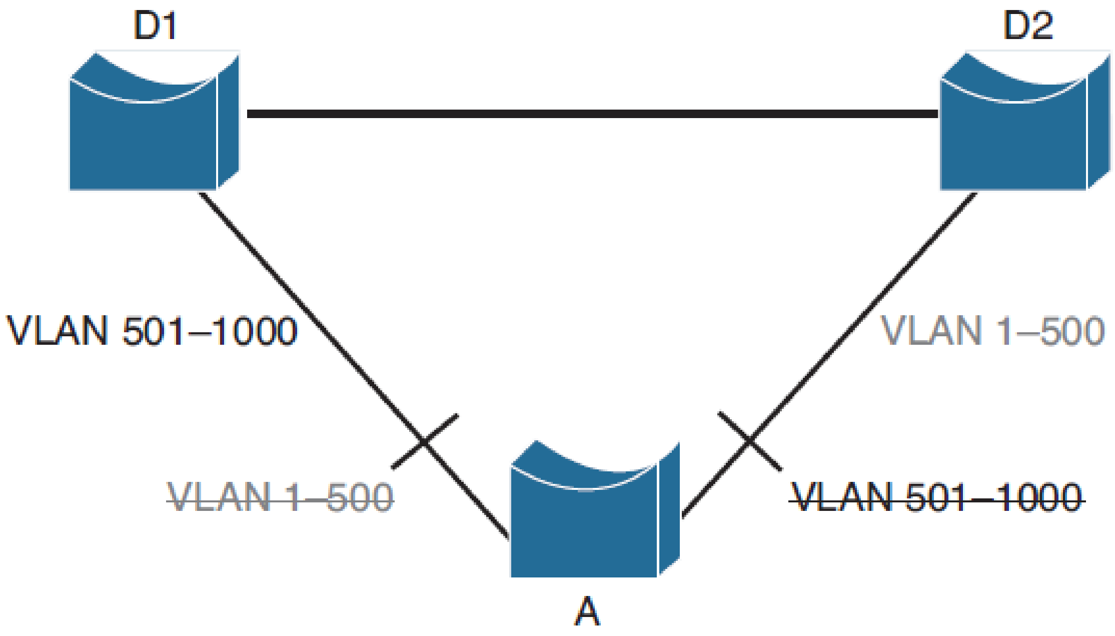


Multiple Spanning Tree Protocol 802.1s

- The main purpose of MST is to reduce the total number of spanning-tree instances to match the physical topology of the network and thus reduce the CPU cycles of a switch
- MST is a concept of mapping one or more VLANs to a single STP instance.
- The number of single STP instances is reduced to the number of links that are available.

VLAN Load Balancing

- Two links and 1000 VLANs
- The 1000 VLANs map to two MST instances. Rather than maintaining 1000 spanning trees, each switch needs to maintain only two spanning trees, reducing the need for switch resources.

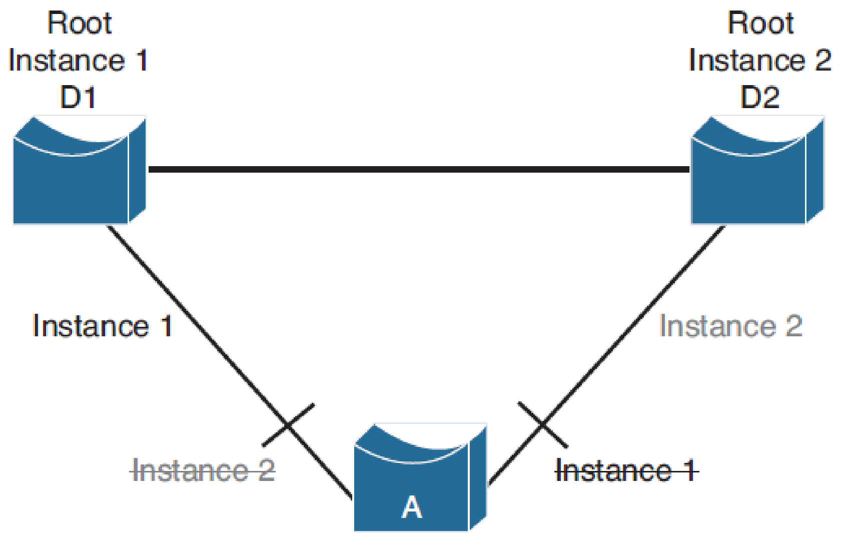


Introducing MST

- MST allows for the building of **multiple spanning trees** over trunks by grouping and associating VLANs to spanning-tree instances.
- Each instance may have a topology that is independent of other spanning-tree instances.
- This architecture provides multiple forwarding paths for data traffic and enables load balancing.
- A failure in one forwarding path does not affect other instances with different forwarding paths; hence, this architecture improves network fault tolerance.
- MST converges faster than PVRST+ and is backward compatible with 802.1D STP, 802.1w (RSTP), and the Cisco PVST+ architecture.

MST

- Benefits
  - Load-balancing scheme is still possible because half the VLANs follow one separate instance.
  - The switch utilization is low because it has to handle only two STP instances.
- Drawbacks
  - The protocol is more complex than the usual spanning tree and therefore requires additional training of the operation staff.
  - Interaction with legacy bridges is sometimes challenging.



MST Configuration Structure

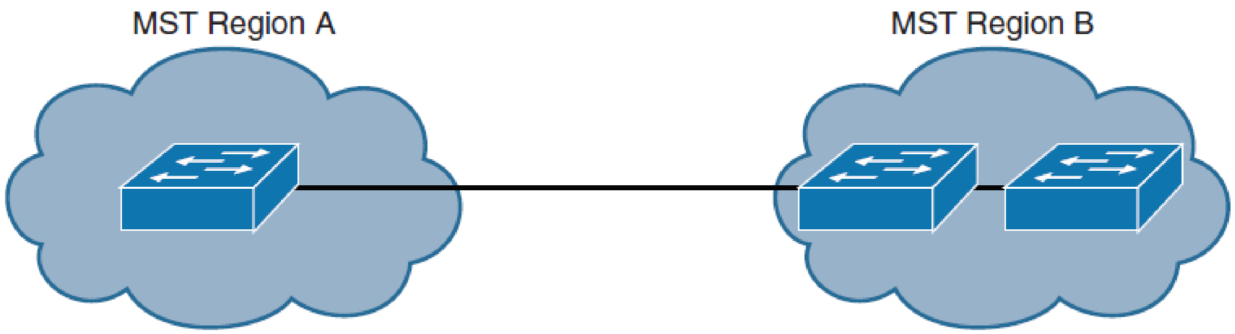
- MST differs from the other spanning-tree implementations in combining some but not necessarily all VLANs into logical spanning-tree instances.
- This difference raises the problem of determining which VLAN is supposed to be associated with which instance.
- VLAN-to-instance association is communicated by tagging the BPDUs so that the receiving device can identify the instances and the VLANs to which they apply.
- To provide this logical assignment of VLANs to spanning trees, each switch that is running MST in the network has a single MST configuration consisting of following three
  - **An alphanumeric configuration name (region)** (32 bytes)
  - **A configuration revision number** (2 bytes)
  - **A 4096-element table** that associates each of the potential 4096 VLANs supported on the chassis with a given instance

MST Configuration Structure

MST Component	Size	Function
MST Configuration Name - identifies an **MST region**	32 bytes	Identifies an MST region. Must be the same across switches in the same region.
Configuration Revision Number	2 bytes	Tracks changes in the MST configuration. Must be updated manually.
VLAN-to-Instance Mapping Table	4096 entries	Maps each VLAN (1-4096) to an MST instance.

MST Regions

- To be part of a common MST region, a group of switches must share the same configuration attributes.
- It is the responsibility of the network administrator to propagate the configuration properly throughout the region.



MST Regions

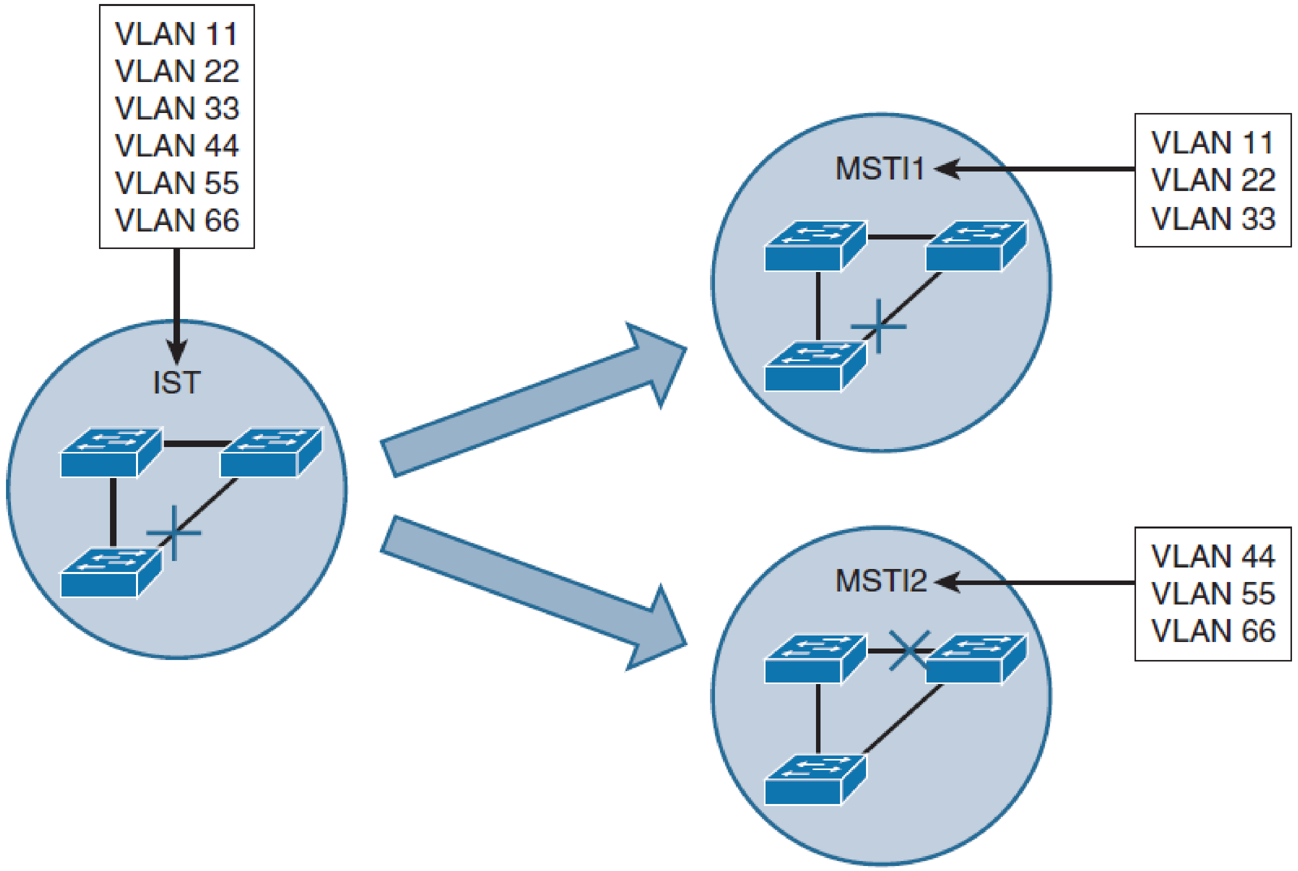
- The exact VLAN-to-instance mapping is not propagated in the BPDU because the switches need to know only whether they are in the same region as a neighbor.
- Therefore, only the **digest** of the VLAN-to-instance mapping table is sent, along with the revision number and the name.
- The digest is a numeric value derived from the VLAN-to-instance mapping table through a mathematical function.
- After a switch receives a BPDU, it extracts the digest and compares it with its own computed digest.
- If the digests differ, the mapping must be different, so the port on which the BPDU was received is at the boundary of a region.

MST Configuration Revision

- The **configuration revision** number gives you a method of tracking changes that are made to the MST region.
- It **does not** automatically increase each time that you make changes to the MST configuration.
- Each time that you make a change, you should increase the revision number by one.

STP Instances with MST

- MST supports a number of instances.
- Instance 0 is the **Common and Internal Spanning Tree** (CIST) aka MSTI0.

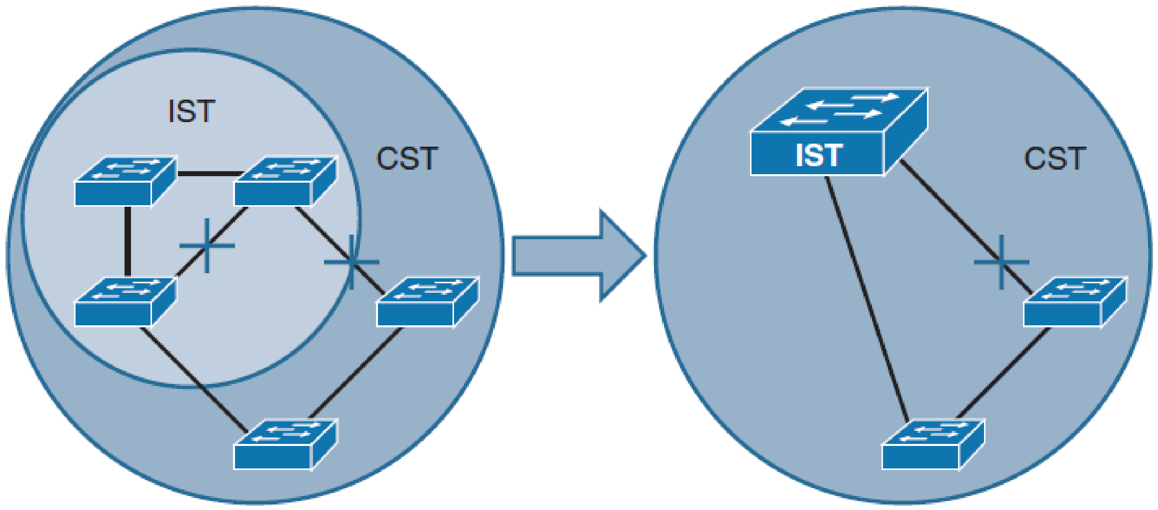


STP Instances with MST

- All six VLAN instances initially belong to MSTI0. This is the default behavior.
- Then make the half of VLAN instances (11, 22, and 33) mapped to MSTI1, and the other half (44, 55, and 66) mapped to MSTI2.
- If different root bridges are configured for MSTI1 and MSTI2, their topologies will converge differently.
- By having different Layer 2 topologies between MST instances, links are more evenly utilized.

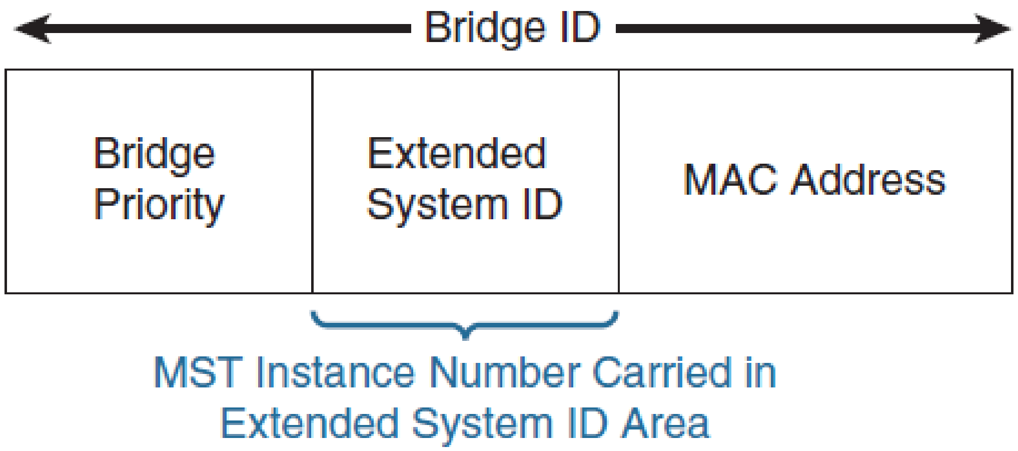
STP Instances with MST

- Within a topology where multiple variations of STP are used, Common Spanning Tree (CST) topology considers an MST region as a single black box.
- CST maintains a loopfree topology with the links that connect the regions to each other and to switches that are not running MST



Extended System ID for MST

- As with PVST, the 12-bit Extended System ID field is used in MST.
- In MST, this field carries the MST instance number.



MST BPDU header fields

- The following fields are **unique to MSTP BPDUs** and **do not exist in standard STP or RSTP BPDUs**:

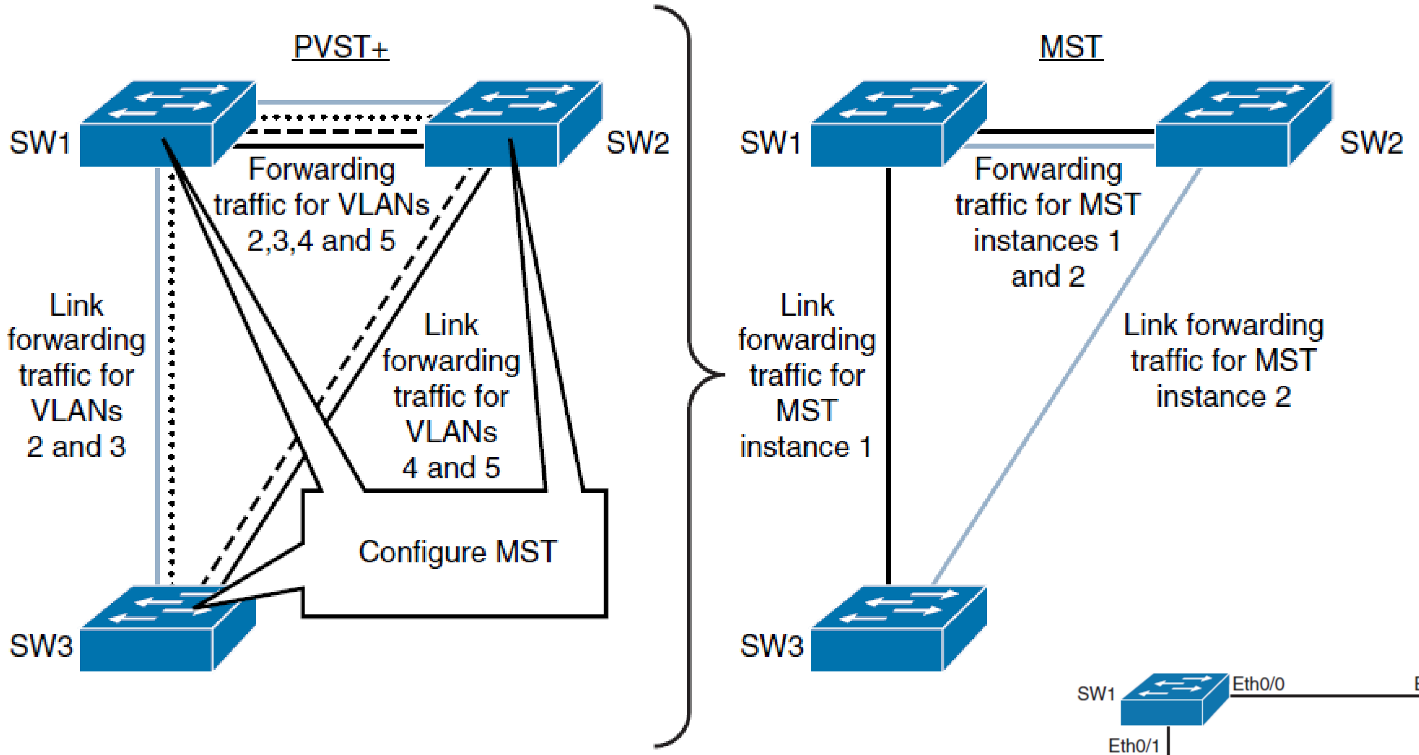
Field	**Size (Bytes)**	**Description**
MST Configuration Identif.	**48 bytes**	Defines the MST region details.
→ **Configuration Name**	**32 bytes**	Name of the MST region (user-defined).
→ **Revision Number**	**2 bytes**	MST configuration version number (set by admin).
→ **Configuration Digest**	**16 bytes**	MD5 hash of the VLAN-to-Instance mapping table
CIST Root Path Cost	4	Cost to reach the CIST root inside the MST region.
CIST Bridge ID	8	Bridge ID of the CIST root switch in the MST region
CIST Remaining Hops	1	TTL (Time To Live) for BPDUs inside MST regions.
MSTI Count	1	Number of MST instances included in the BPDU.
MSTI Records (per MSTI)	**(Variable)**	Includes data for each MST instance
→ **MSTI Root ID**	8	Root Bridge ID for this MSTI.
→ **MSTI Bridge Priority**	2	Bridge priority for this MSTI.
→ **MSTI Port Priority**	2	Port priority for this MSTI.
→ **MSTI Remaining Hops**	1	Similar to TTL, used inside MST regions.

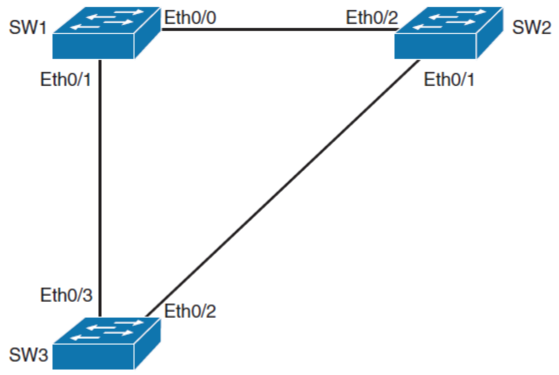
MST BPDU header fields

- The BPDU will look like this:

```
+-------------------------------------------------------+
| Standard BPDU Header (CIST)                           |
| → Protocol ID: 0x0000                                 |
| → Version: 0x03 (MSTP)                                |
| → BPDU Type: 0x02                                     |
| → Root Bridge ID: 32768:aaaa.bbbb.cccc                |
| → Root Path Cost: 0                                   |
+-------------------------------------------------------+
| MST Configuration Identifier                          |
| → Name: "MST-Region1"                                 |
| → Revision: 1                                         |
| → MD5 Hash: (Checksum)                                |
+-------------------------------------------------------+
| MSTI 1 Information (VLANs 10-20)                      |
| → MSTI Root ID: 32768:xxxx.yyyy.zzzz                  |
| → Bridge Priority: 32768                              |
| → Port Priority: 128                                  |
| → Remaining Hops: 20                                  |
+-------------------------------------------------------+
| MSTI 2 Information (VLANs 30-40)                      |
| → MSTI Root ID: 32768:wwww.vvvv.uuuu                  |
| → Bridge Priority: 32768                              |
| → Port Priority: 128                                  |
| → Remaining Hops: 20                                  |
+-------------------------------------------------------+
```

Configuring and Verifying MST


PVST+
SW1
SW2
SW3
Forwarding traffic for VLANs 2,3,4 and 5
Link forwarding traffic for VLANs 2 and 3
Link forwarding traffic for VLANs 4 and 5
Configure MST
MST
SW1
SW2
SW3
Forwarding traffic for MST instances 1 and 2
Link forwarding traffic for MST instance 1
Link forwarding traffic for MST instance 2



Configuring MST with the CCNP Region

```
SW1(config)# spanning-tree mst configuration
SW1(config-mst)# name CCNP
SW1(config-mst)# revision 1

SW2(config)# spanning-tree mst configuration
SW2(config-mst)# name CCNP
SW2(config-mst)# revision 1

SW3(config)# spanning-tree mst configuration
SW3(config-mst)# name CCNP
SW3(config-mst)# revision 1
```

MST Instance 1 and 2 Configuration

```
SW1(config)# spanning-tree mst configuration
SW1(config-mst)# instance 1 vlan 2,3
SW1(config-mst)# instance 2 vlan 4,5
SW1(config-mst)# end

SW2(config)# spanning-tree mst configuration
SW2(config-mst)# instance 1 vlan 2,3
SW2(config-mst)# instance 2 vlan 4,5
SW2(config-mst)# end

SW3(config)# spanning-tree mst configuration
SW3(config-mst)# instance 1 vlan 2,3
SW3(config-mst)# instance 2 vlan 4,5
SW3(config-mst)# end
```

- MST is configured with three instances:
  - VLANs 2 and 3 belong to **instance 1**
  - VLANs 4 and 5 belong to **instance 2**
- All other VLANs between 1 and 4094, that are not in instances 1 or 2, belong to instance 0.

SPT Root Bridge Configuration

```
SW1(config)# spanning-tree mst 1 root primary
SW1(config)# spanning-tree mst 2 root secondary
```

```
SW2(config)# spanning-tree mst 1 root secondary
SW2(config)# spanning-tree mst 2 root primary

Change STP mode to MST on all three switches as shown in example 4-29.
Example 4-29 Changing the SPT mode to MST
SW1(config)# spanning-tree mode mst


SW2(config)# spanning-tree mode mst


SW3(config)# spanning-tree mode mst
```

- Changing STP mode to MST before doing the actual VLAN-to-instance mappings **is not advisable**. Every change in the mapping will result in recalculation of the STP tree.
- A switch cannot run MST and PVST+ at the same time.

Verifying MST

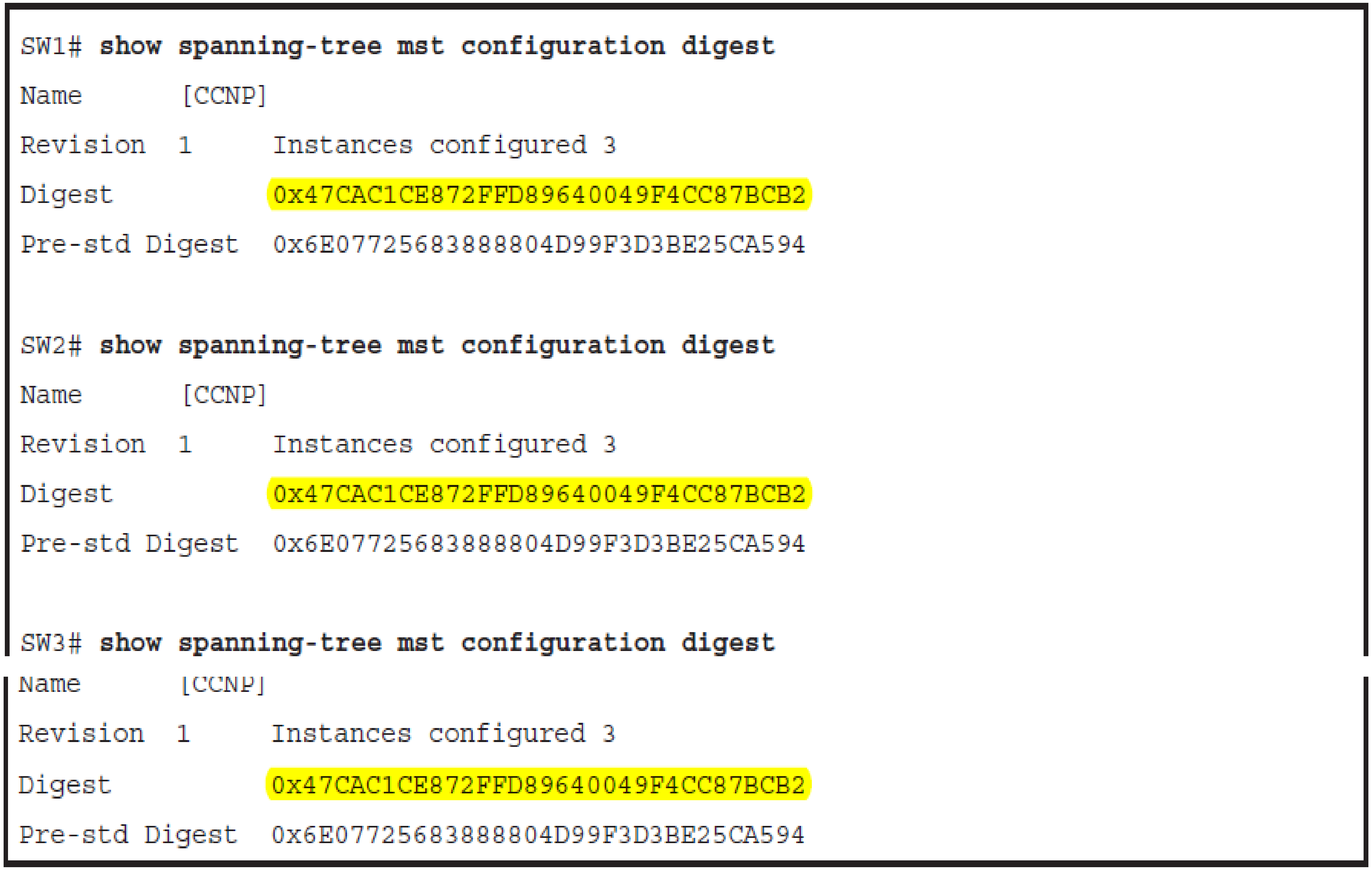
```
SW3# show spanning-tree summary
Switch is in mst mode (IEEE Standard)
<... output omitted ...>

Name       Blocking Listening Learning Forwarding STP Active
---------- -------- --------- -------- ---------- ----------
MST0         0        0         0        24         24
MST1         0        0         0         4          4
MST2         0        0         0         4          4
--------- -------- --------- -------- ---------- ----------
3 msts       0        0         0        32         32
```

```
SW3(config)# spanning-tree mst configuration
SW3(config-mst)# show current
Current MST configuration
Name      [CCNP]
Revision  1     Instances configured 3

Instance  Vlans mapped
--------  -----------------------------------------------------------
0         1,6-4094
1         2-3
2         4-5
---------------------------------------------------------------------
```

Verifying MST Digest

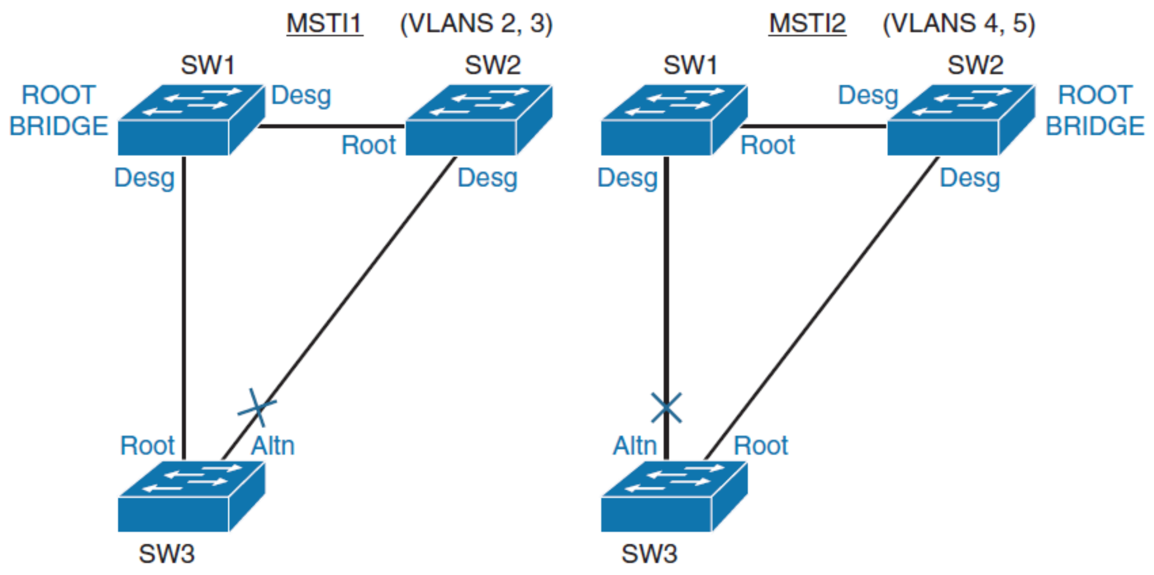


```
SW1# show spanning-tree mst configuration digest
Name          [CCNP]
Revision   1       Instances configured 3
Digest             0x47CAC1CE872FFD89640049F4CC87BCB2
Pre-std Digest     0x6E077256838888O4D99F3D3BE25CA594

SW2# show spanning-tree mst configuration digest
Name          [CCNP]
Revision   1       Instances configured 3
Digest             0x47CAC1CE872FFD89640049F4CC87BCB2
Pre-std Digest     0x6E077256838888O4D99F3D3BE25CA594

SW3# show spanning-tree mst configuration digest
Name          [CCNP]
Revision   1       Instances configured 3
Digest             0x47CAC1CE872FFD89640049F4CC87BCB2
Pre-std Digest     0x6E077256838888O4D99F3D3BE25CA594
```

Verifying MST Instances Mappings



```
SW3# show spanning-tree mst 1

##### MST1    vlans mapped:   2-3
<... output omitted ..>
Et0/2               Altn BLK 2000000   128.3    Shr
Et0/3               Root FWD 2000000   128.4    Shr
<... output omitted ..>

SW3# show spanning-tree mst 2

##### MST2    vlans mapped:   4-5
<... output omitted ..>
Et0/2               Root FWD 2000000   128.3    Shr
Et0/3               Altn BLK 2000000   128.4    Shr
<... output omitted ..>
```

Configuring MST Path Cost

- Path cost functions the same as with other STPs, except with MST port costs are configured per instance.

```
Switch(config)# interface Ethernet 0/2
Switch(config-if)# spanning-tree mst 1 cost 1000000
! Sets the MST cost of the interface to 1000000
SW3# show spanning-tree mst
<... output omitted ..>
Interface          Role   Sts  Cost    Prio.Nbr  Type
------------------ ----   ---- ------- --------- -----
Et0/2              Altn   BLK  1000000 128.3     Shr
Et0/3              Root   FWD  2000000 128.4     Shr
Et1/0              Desg   FWD  2000000 128.5     Shr
Et1/1              Desg   FWD  2000000 128.6     Shr
! Verify MST path cost configuration
```

Configuring MST Port Priority

- Port priority functions the same as with other STPs, except with MST port priorities are configured per instance.

```
Switch(config)# interface Ethernet 0/2
Switch(config-if)# spanning-tree mst 1 port-priority 32
! Sets the MST port priority for given MST interface

SW3# show spanning-tree mst
<... output omitted ..>
Interface           Role   Sts  Cost     Prio.Nbr  Type
------------------- ----   ---- -------- --------- -----
Et0/2               Altn   BLK  1000000   32.3     Shr
Et0/3               Root   FWD  2000000  128.4     Shr
Et1/0               Desg   FWD  2000000  128.5     Shr
Et1/1               Desg   FWD  2000000  128.6     Shr
! Verify port ID settings that are sent
```

MST Protocol Migration

- Ensure that all switch-to-switch links on which a rapid transition is desired are full duplex.
- Edge ports are defined through the PortFast feature.
- Carefully decide how many instances are needed in the switched network, and keep in mind that an instance translates to a logical topology.
- Decide what VLANs to map onto those instances, and carefully select a root and a backup root for each instance.
- Choose a configuration name and a revision number that will be common to all switches in the network.
  - Cisco recommends that you place as many switches as possible into a single region; it is not advantageous to segment a network into separate regions.

MST Protocol Migration

- Avoid mapping any VLANs onto instance 0.
- Migrate the core first. Change the STP type to MST and work your way down to the access switches.
- The configuration of the features such as the PortFast, BPDU Guard, BPDUF Filter, Root Guard, and Loop Guard are also applicable in MST mode.
- If you have already enabled these features in the PVST+ mode, it remains active after the migration to MST mode.

Chapter 4 Summary

- Spanning Tree Protocol (STP) overview, its operations, and history
- Implement Rapid Spanning Tree Protocol (RSTP)
- Describe how and where to configure the following features: PortFast, UplinkFast, BackboneFast, BPDU Guard, BPDU Filter, Root Guard, Loop Guard, Unidirectional Link Detection, and FlexLinks
- Configure Multiple Spanning Tree (MST)
- Troubleshooting STP

Chapter 4 Labs

- **CCNPv7.1 SWITCH Lab4.1 STP**
- **CCNPv7.1 SWITCH Lab4.2 MST**

Cisco | Networking Academy®
Mind Wide Open™

Acknowledgment

- *Some of the images and texts are from Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide: (CCNP SWITCH 300-115)* by Richard Froom and Erum Frahim (1587206641)
- Copyright © 2015 – 2016 Cisco Systems, Inc.
- Special Thanks to *Bruno Silva*