Spanning Tree Protocol

Md. Abdul Awal
awal.ece@gmail.com

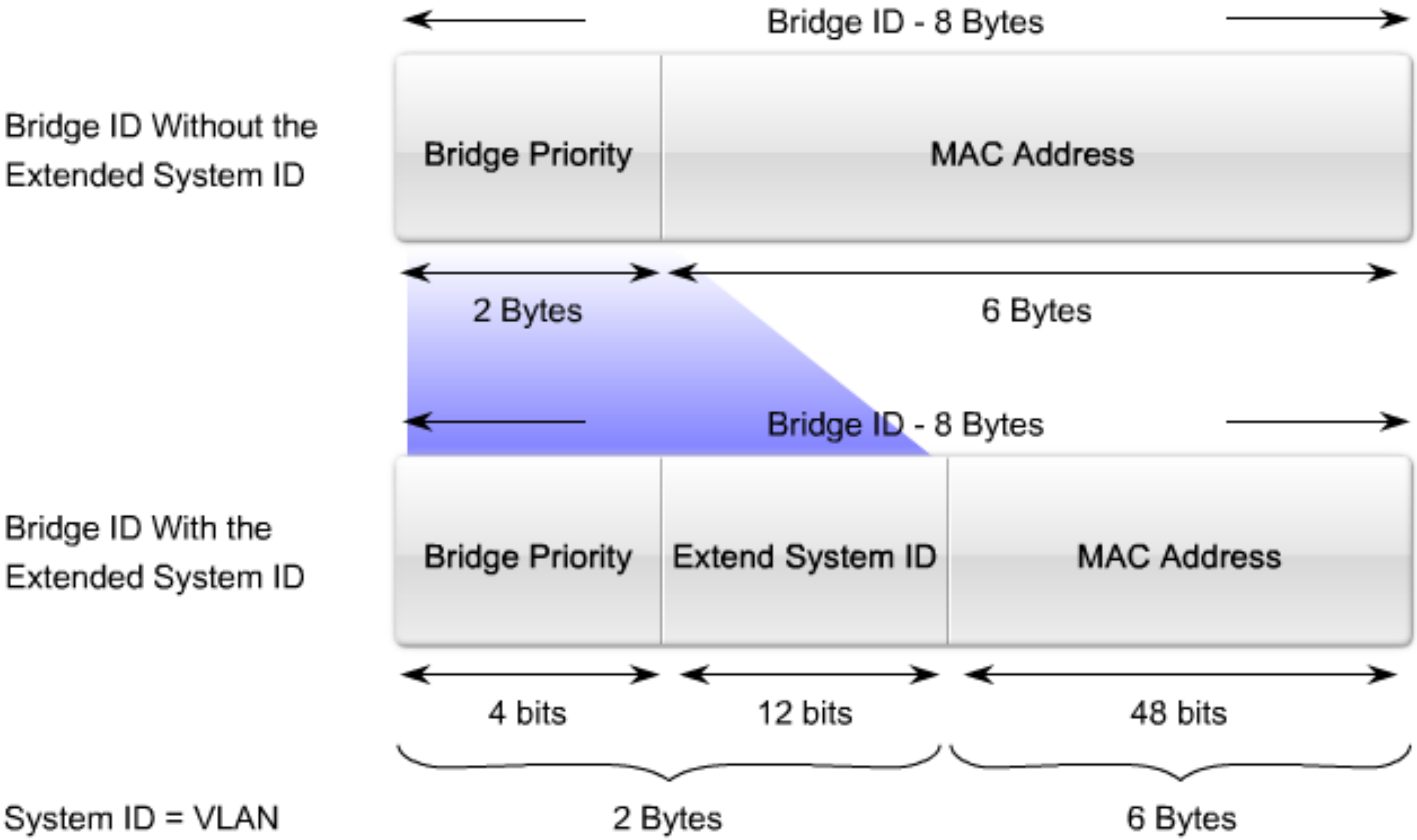
STP Operation

- Switches exchange BPDU frames to determine which switch has the lowest bridge ID (BID)
- The switch with lowest BID becomes the root bridge
- STA calculates the shortest path to the root bridge
- STA considers both path and port costs when determining which path to leave unblocked
- Each switch maintains local information about its own BID, the root ID, and the path cost to the root

STP Operation

- Every spanning-tree instance (switched LAN or broadcast domain) has a switch designated as the root bridge
- An election process determines which switch becomes the root bridge
- All switches in the broadcast domain participate in the election process
- By default, BPDU frames are sent every 2 seconds after a switch is booted

Bridge ID (BID)



BID Fields

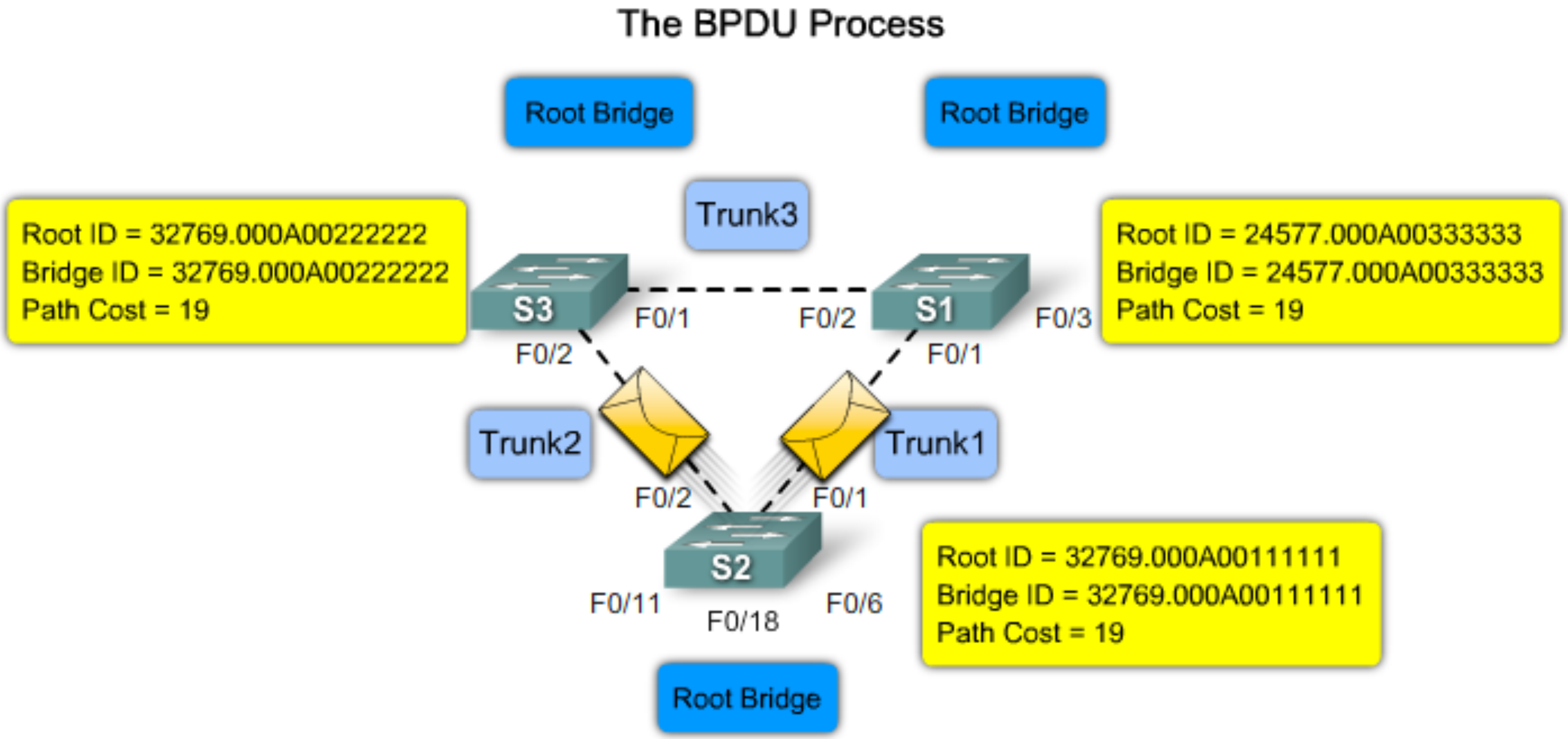
- Bridge Priority
  - Customizable between 1 to 65536, cisco default is 32768
  - The switch with the lowest priority, which means lowest BID, becomes the root bridge
- Extended System ID
  - Generally contains VLAN ID
  - Ranges from 1 to 4096. Therefore, bridge priority values can only be multiples of 4096
- MAC Address
  - If all switches have the same priority and same VLAN ID, switch with lowest MAC address becomes root bridge

Port Cost

Link Speed	Cost
10 Gbps	2
1 Gbps	4
100 Mbps	19
10 Mbps	100

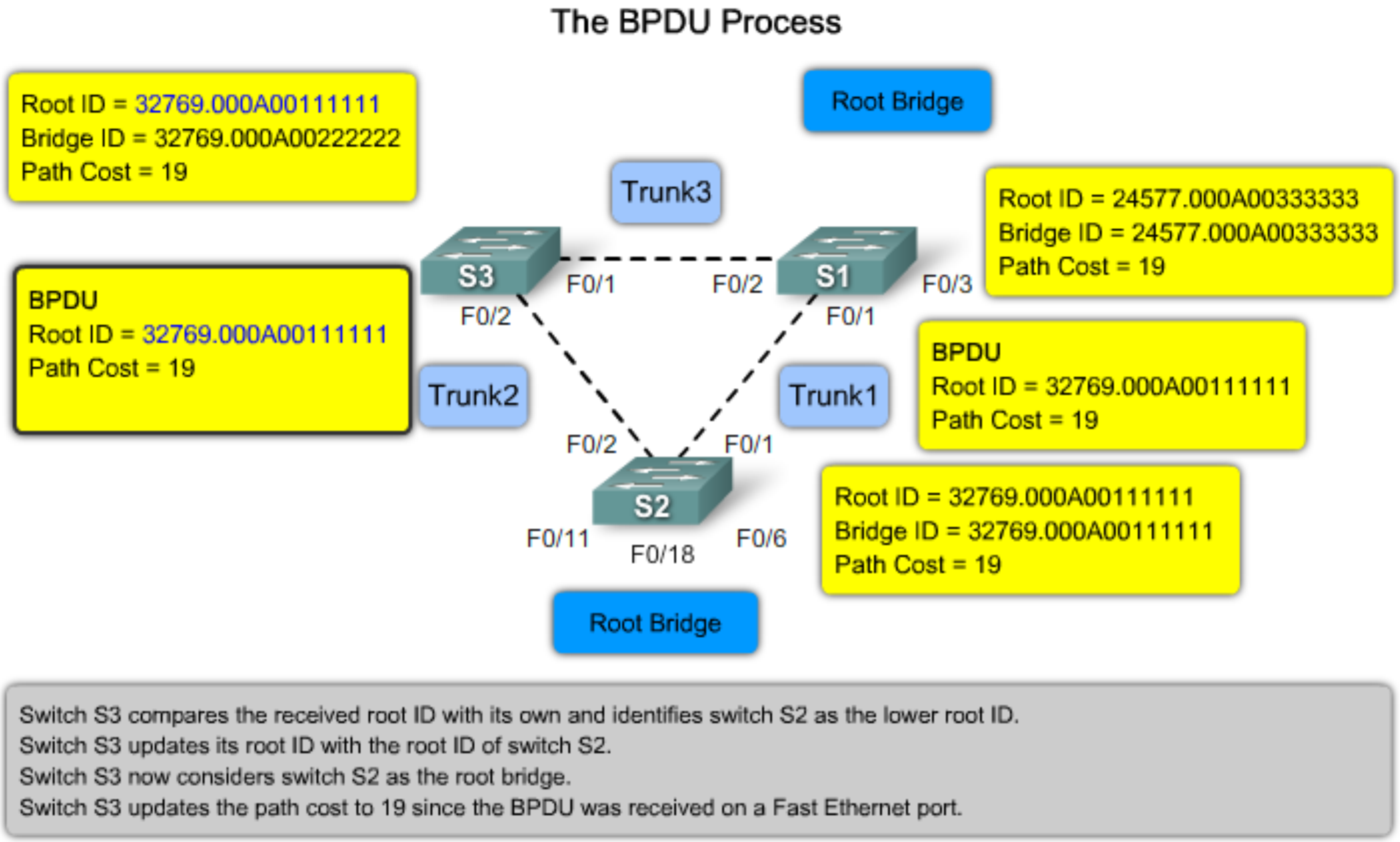
- Port Cost is configurable
- Path cost is the sum of all the port costs along the path to the root bridge
- The paths with the lowest path cost become the preferred path, and all other redundant paths are blocked

BPDU Process


The BPDU Process
Root Bridge
Root Bridge
Trunk3
Root ID = 32769.000A00222222
Bridge ID = 32769.000A00222222
Path Cost = 19
S3
F0/1
F0/2
F0/2
S1
F0/3
F0/1
Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19
Trunk2
Trunk1
F0/2
F0/1
S2
F0/11
F0/18
F0/6
Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19
Root Bridge

Switch S2 forwards BPDU frames out of all switch ports. The BPDU frame contains the bridge ID and the root ID of Switch S2 indicating that it is the root bridge.

BPDU Process



The BPDU Process

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00222222
Path Cost = 19

BPDU
Root ID = 32769.000A00111111
Path Cost = 19

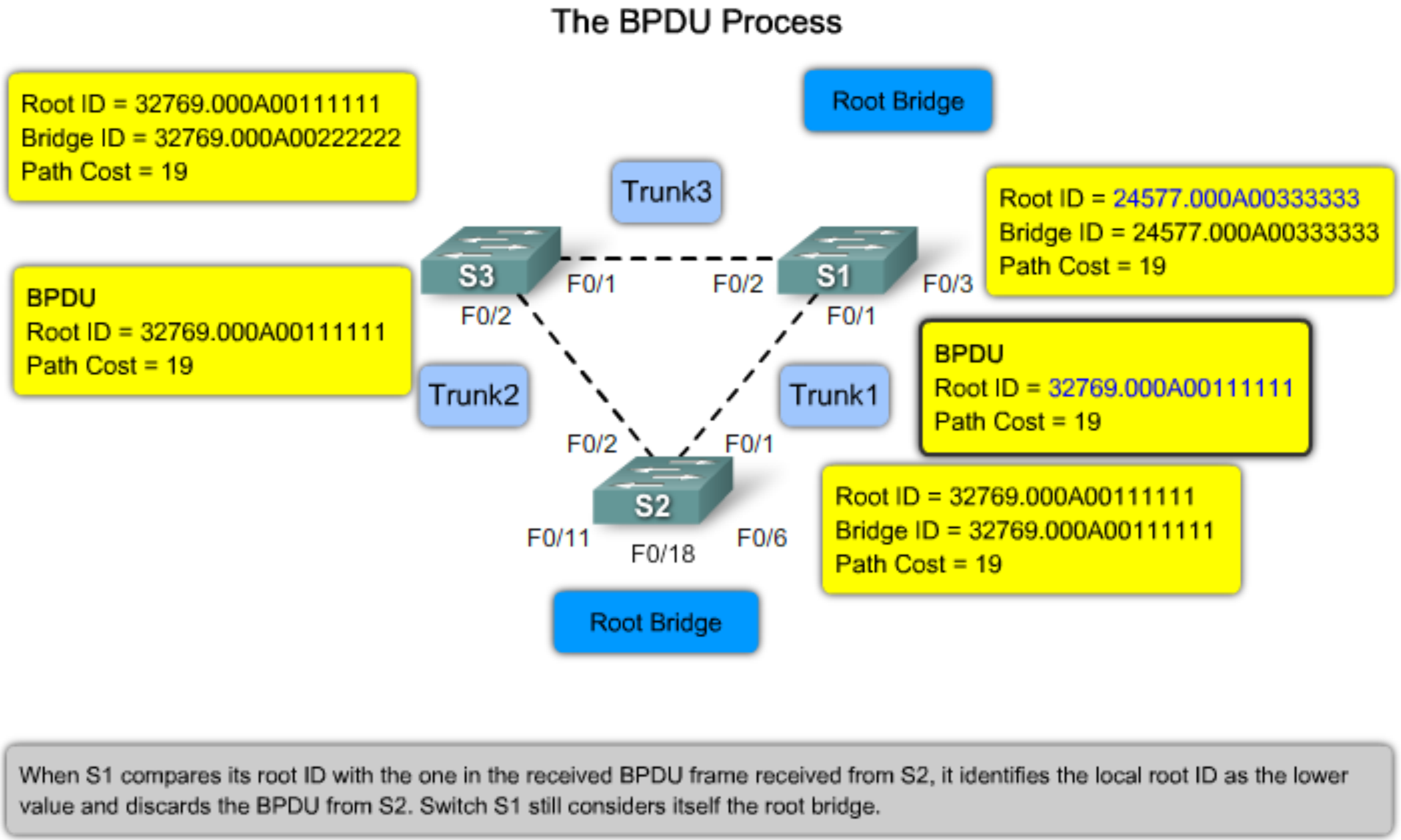
Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19

BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19

Switch S3 compares the received root ID with its own and identifies switch S2 as the lower root ID.
Switch S3 updates its root ID with the root ID of switch S2.
Switch S3 now considers switch S2 as the root bridge.
Switch S3 updates the path cost to 19 since the BPDU was received on a Fast Ethernet port.

BPDU Process



The BPDU Process

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00222222
Path Cost = 19

BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Root Bridge

Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19

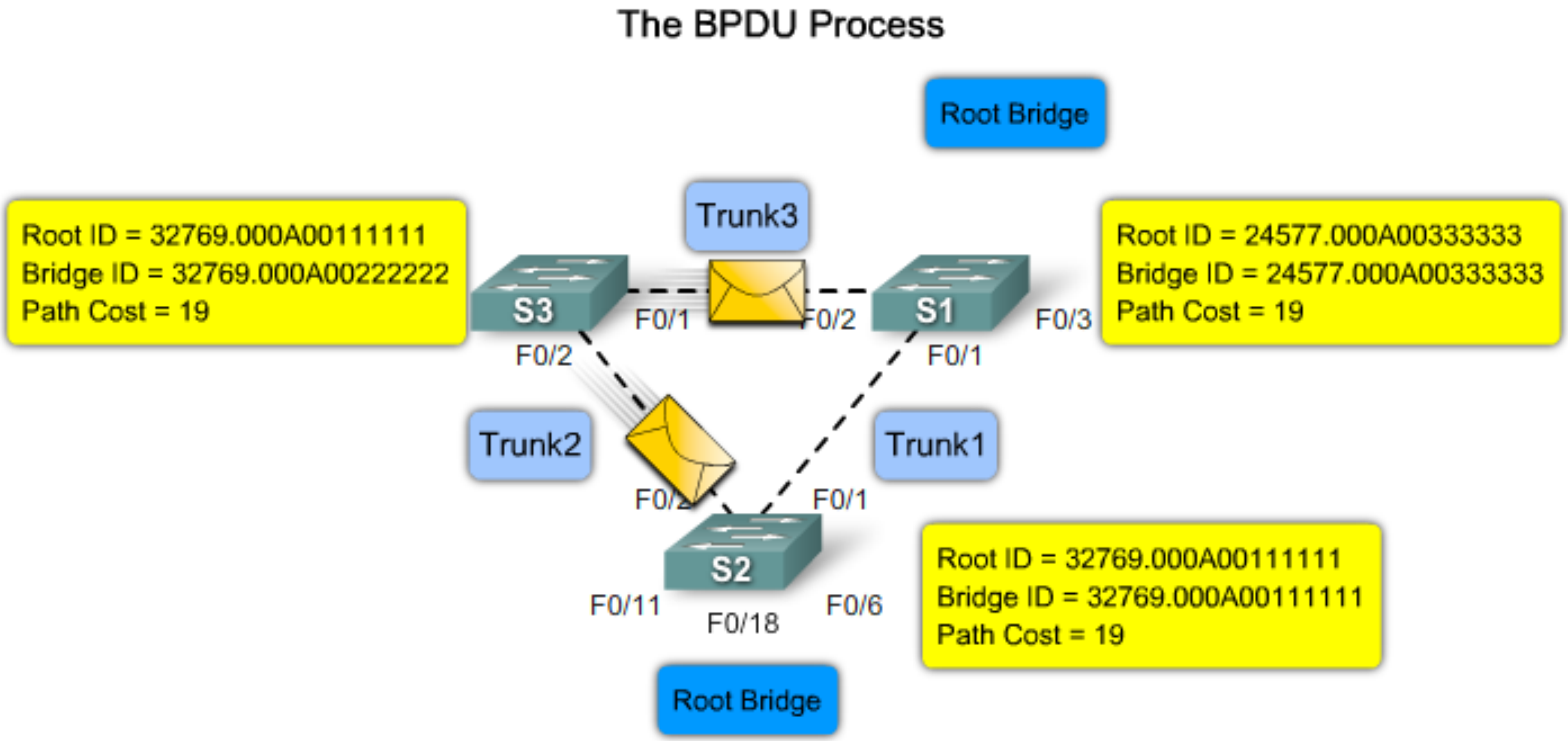
BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19

Root Bridge

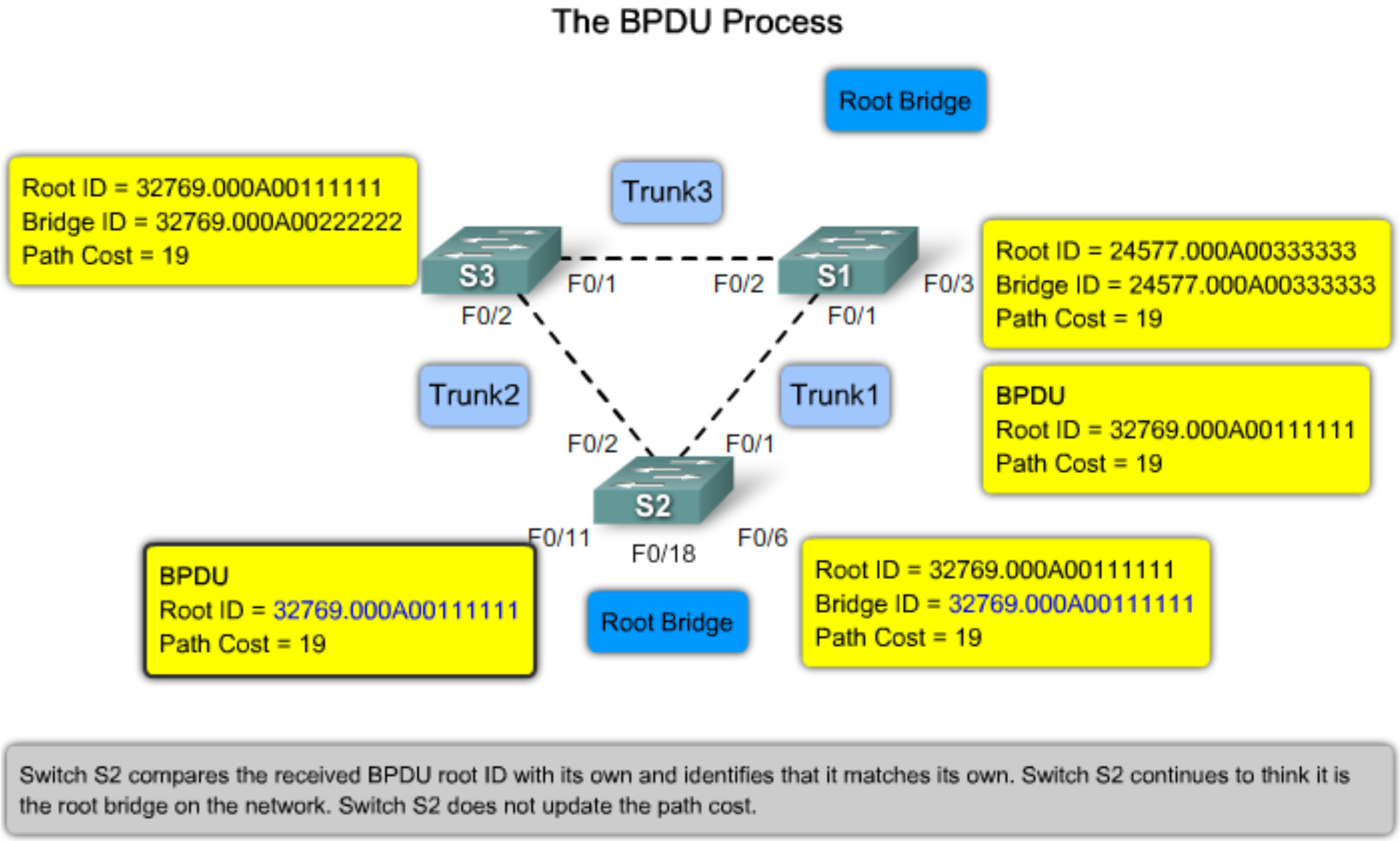
When S1 compares its root ID with the one in the received BPDU frame received from S2, it identifies the local root ID as the lower value and discards the BPDU from S2. Switch S1 still considers itself the root bridge.

BPDU Process


The BPDU Process
Root Bridge
Trunk3
Root ID = 32769.000A00111111
Bridge ID = 32769.000A00222222
Path Cost = 19
S3
F0/1
F0/2
S1
F0/3
Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19
F0/2
F0/1
Trunk2
Trunk1
F0/2
F0/1
S2
F0/11
F0/18
F0/6
Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19
Root Bridge

Switch S3 forwards BPDU frames out of all switch ports. The BPDU frame contains the root ID of Switch S2 indicating that it is the root bridge.

BPDU Process



The BPDU Process

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00222222
Path Cost = 19

Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19

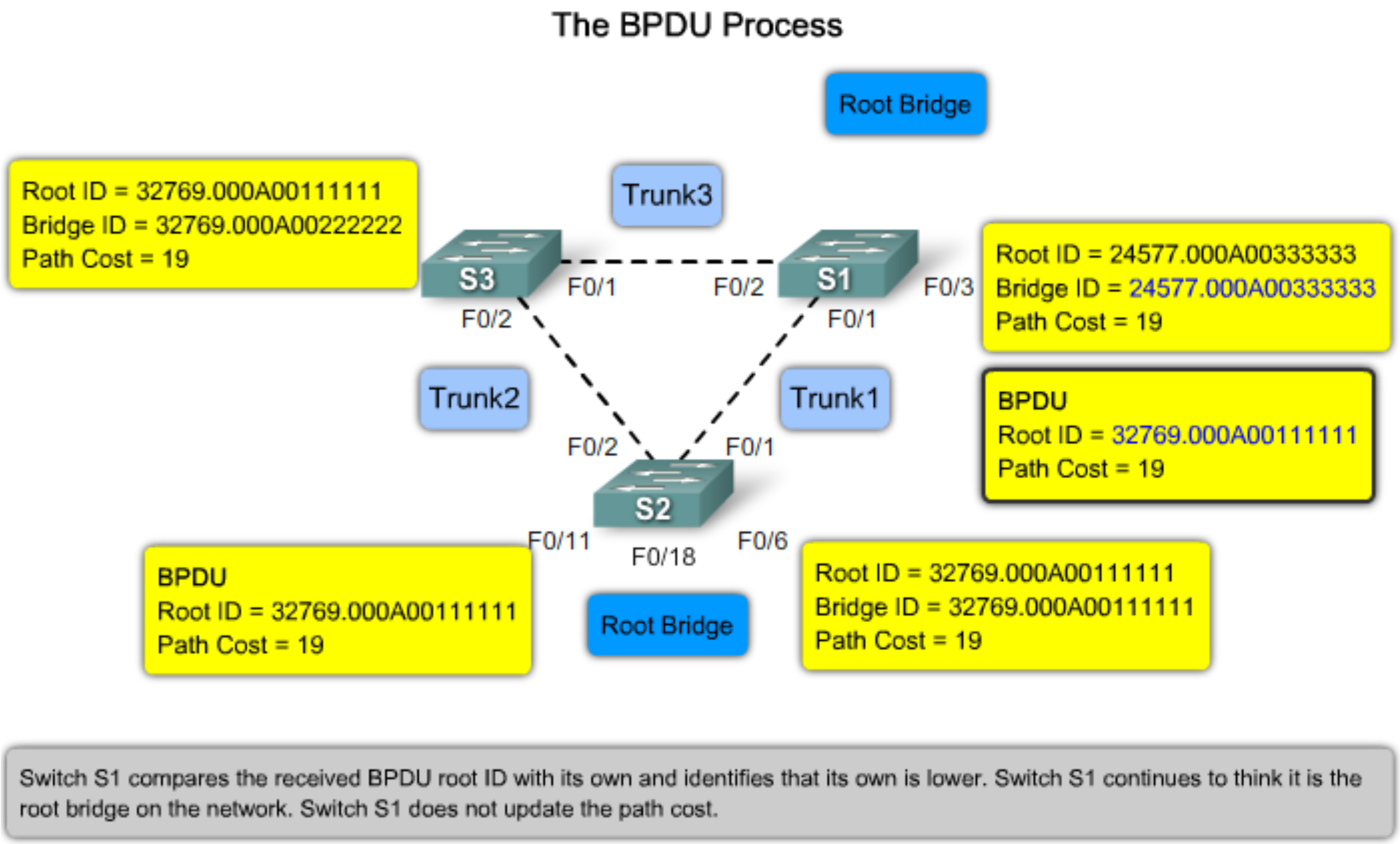
BPDU
Root ID = 32769.000A00111111
Path Cost = 19

BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19

Switch S2 compares the received BPDU root ID with its own and identifies that it matches its own. Switch S2 continues to think it is the root bridge on the network. Switch S2 does not update the path cost.

BPDU Process



The BPDU Process

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00222222
Path Cost = 19

Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19

BPDU
Root ID = 32769.000A00111111
Path Cost = 19

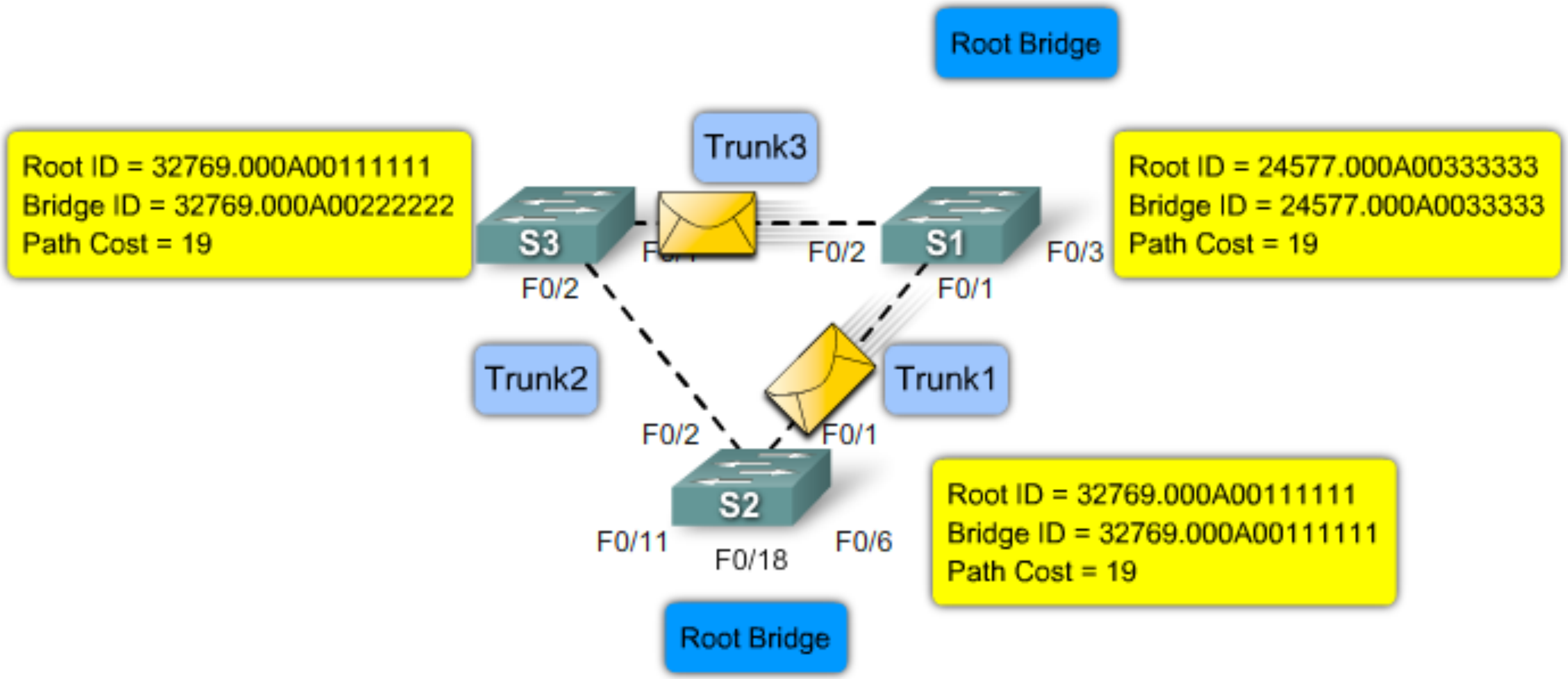
BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19

Switch S1 compares the received BPDU root ID with its own and identifies that its own is lower. Switch S1 continues to think it is the root bridge on the network. Switch S1 does not update the path cost.

BPDU Process

The BPDU Process



Root ID = 32769.000A00111111
Bridge ID = 32769.000A00222222
Path Cost = 19

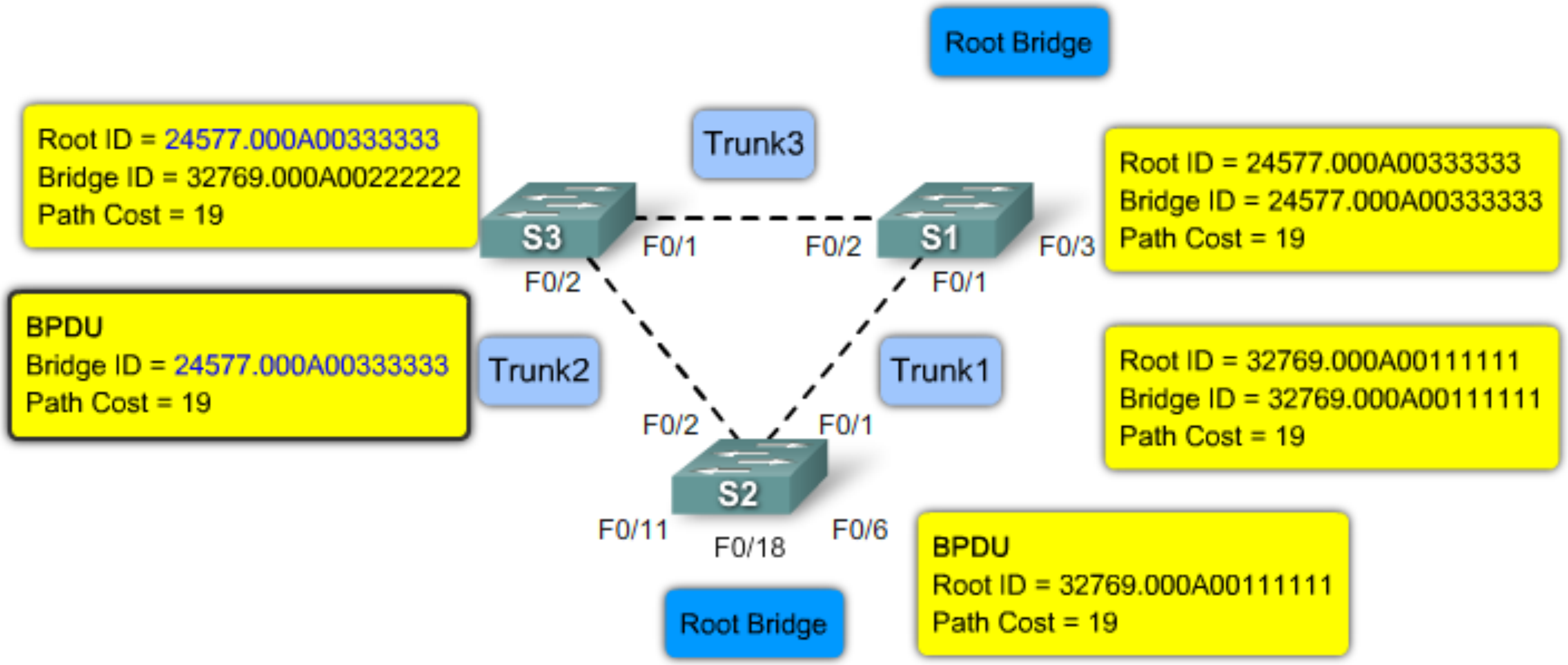
Root ID = 24577.000A00333333
Bridge ID = 24577.000A0033333
Path Cost = 19

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19

Switch S1 forwards BPDU frames out of all switch ports. The BPDU frame contains the bridge ID and root ID of Switch S1 indicating that it is the root bridge.

BPDU Process

The BPDU Process



Root ID = 24577.000A00333333
Bridge ID = 32769.000A00222222
Path Cost = 19

BPDU
Bridge ID = 24577.000A00333333
Path Cost = 19

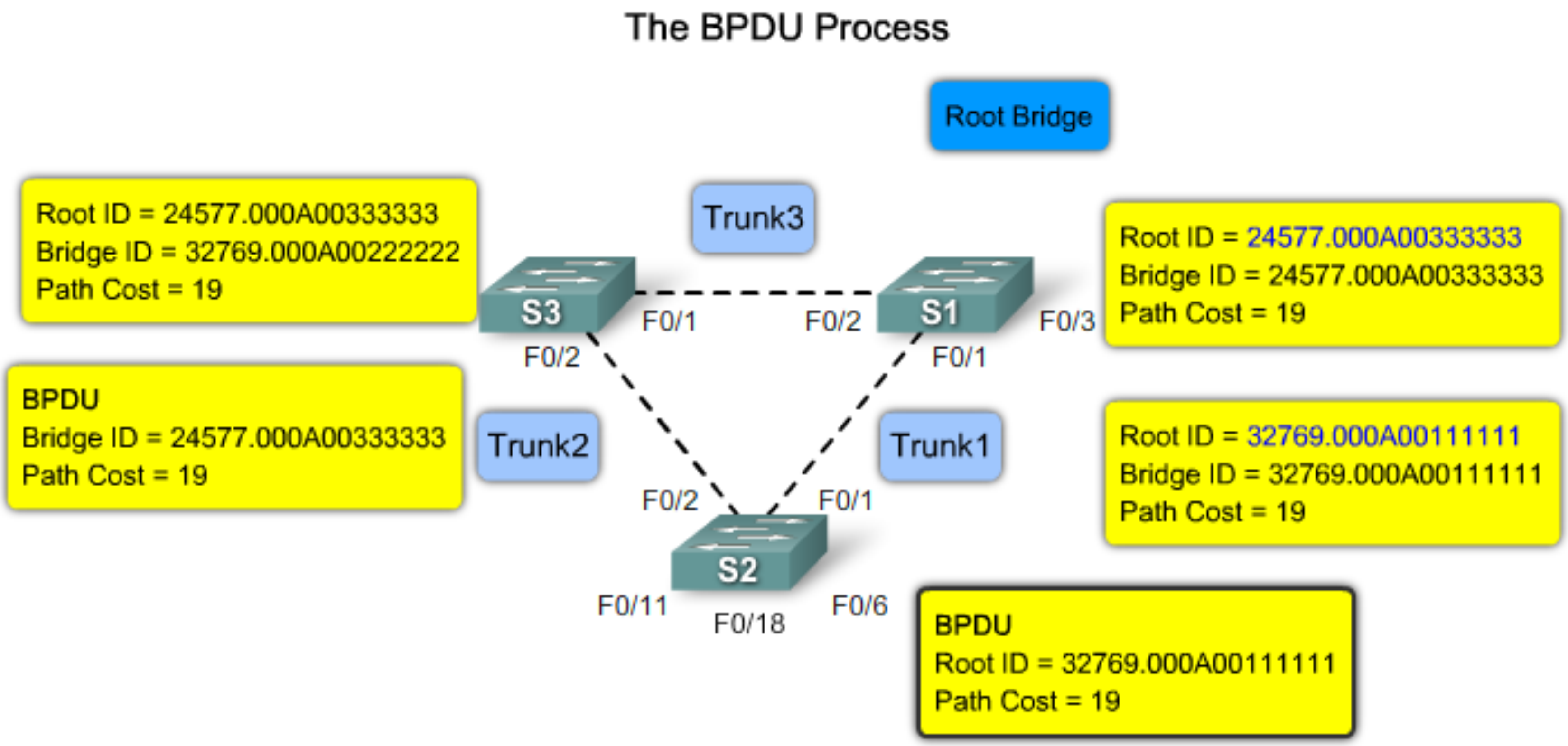
Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19

Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19

BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Switch S3 compares the received root ID with its own and identifies switch S1 as the lower root ID. Switch S3 updates its root ID with the root ID of switch S1. Switch S3 now considers switch S1 as the root bridge. Switch S3 updates the path cost to 19 since the BPDU was received on a Fast Ethernet port.

BPDU Process

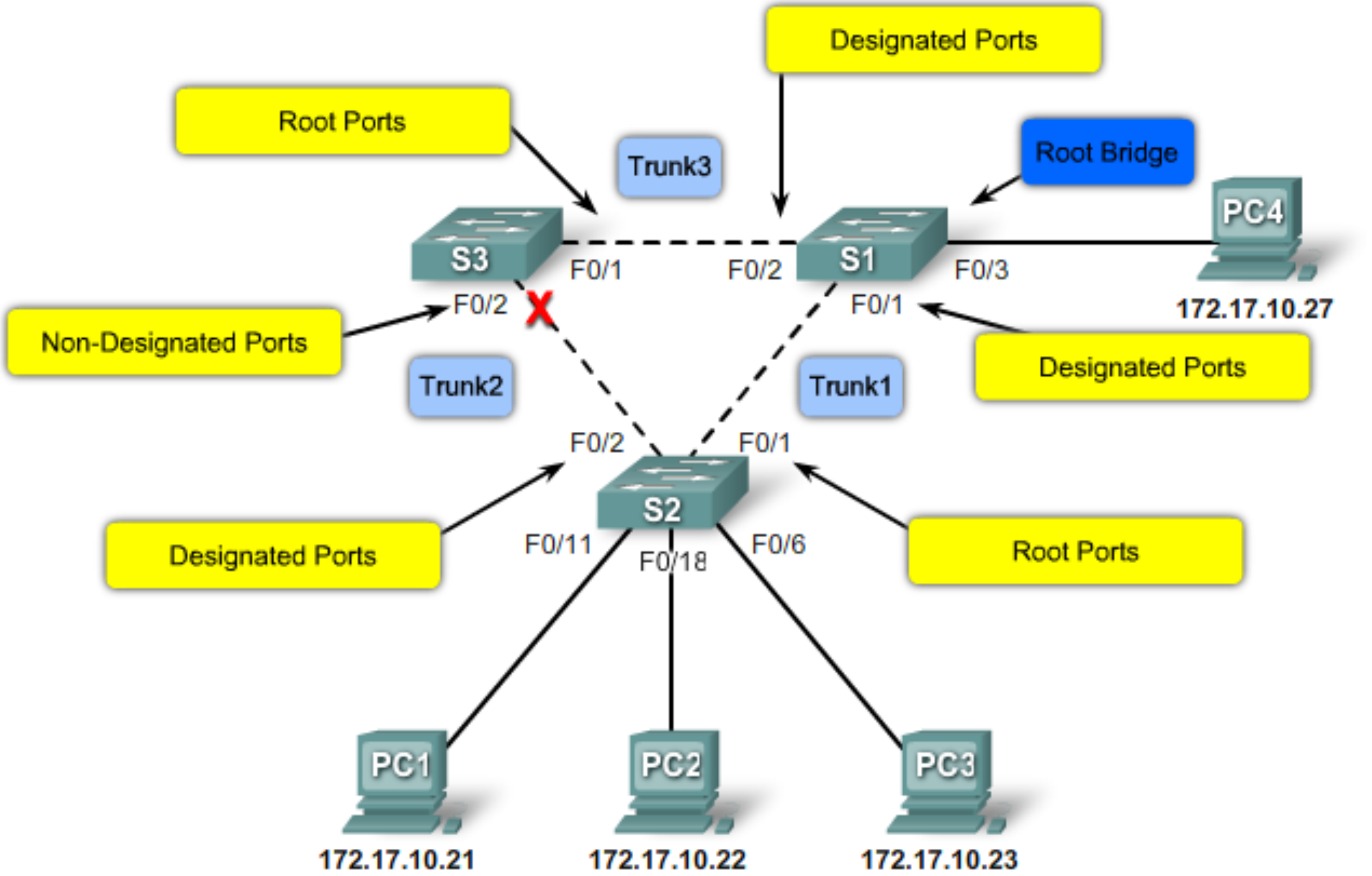

The BPDU Process
Root Bridge
Root ID = 24577.000A00333333
Bridge ID = 32769.000A00222222
Path Cost = 19
Trunk3
S3
F0/1
F0/2
S1
F0/3
Root ID = 24577.000A00333333
Bridge ID = 24577.000A00333333
Path Cost = 19
F0/2
F0/1
BPDU
Bridge ID = 24577.000A00333333
Path Cost = 19
Trunk2
Trunk1
Root ID = 32769.000A00111111
Bridge ID = 32769.000A00111111
Path Cost = 19
F0/2
F0/1
S2
F0/11
F0/18
F0/6
BPDU
Root ID = 32769.000A00111111
Path Cost = 19

Switch S2 compares the received root ID with its own and identifies switch S1 as the lower root ID. Switch S2 updates its root ID with the root ID of switch S1. Switch S2 now considers switch S1 as the root bridge. Switch S2 updates the path cost to 19 since the BPDU was received on a Fast Ethernet port.

Port Roles

- **Root ports -** The root port exists on non-root bridges and is the switch port with the best path to the root bridge. Root ports forward traffic toward the root bridge. Root ports are capable of populating the MAC table
- **Designated ports -** The designated port exists on root and non-root bridges. For root bridges, all switch ports are designated ports. For non-root bridges, a designated port is the switch port that receives and forwards frames toward the root bridge as needed. Only one designated port is allowed per segment. Designated ports are capable of populating the MAC table
- **Non-designated ports -** The non-designated port is a switch port that is blocked, so it is not forwarding data frames and not populating the MAC address table with source addresses. A non-designated port is not a root port or a designated port
- **Disabled Port -** The disabled port is a switch port that is administratively shut down. A disabled port does not function in the spanning-tree process

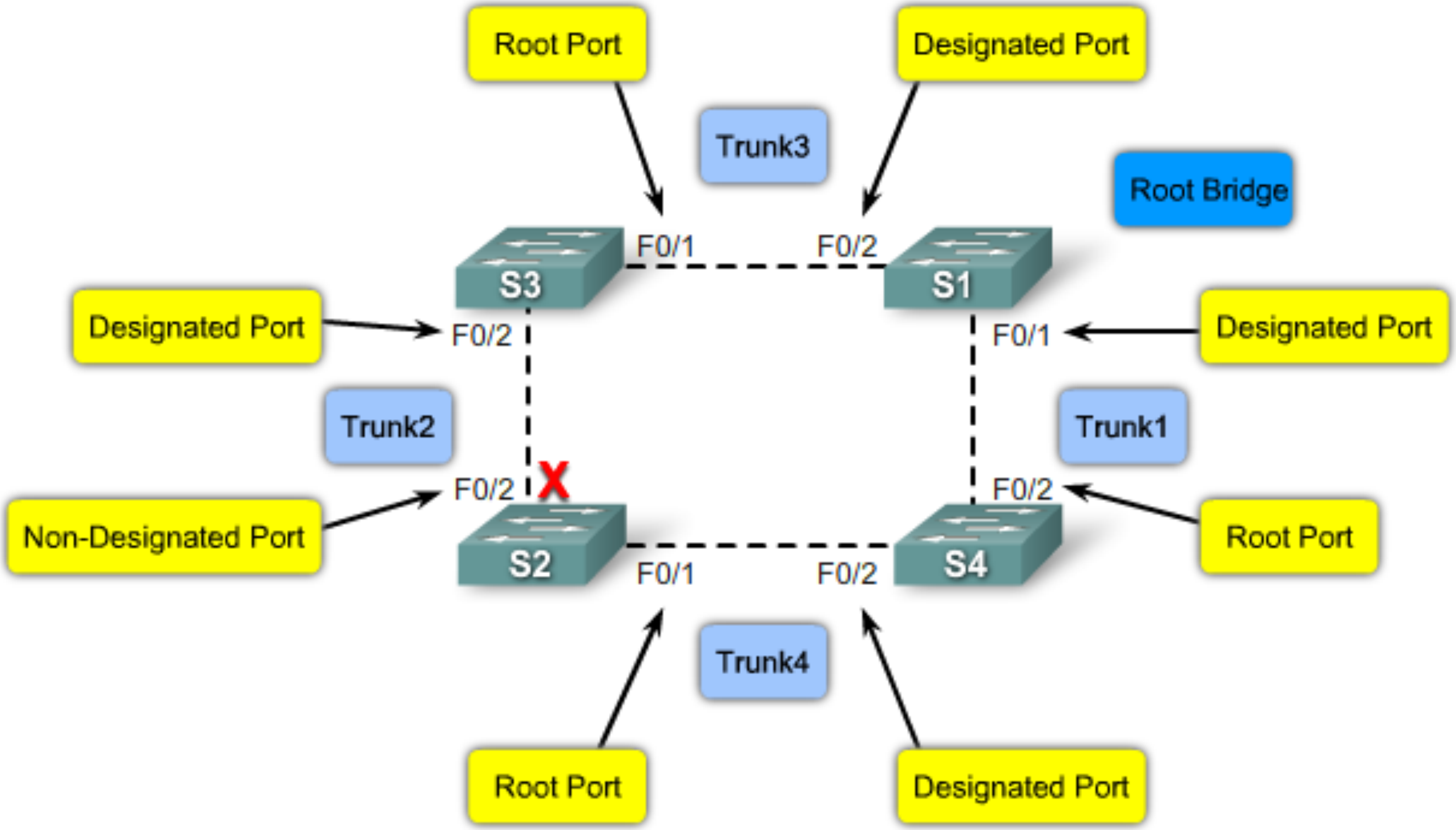
Port Roles


Designated Ports
Root Ports
Trunk3
Root Bridge
PC4
S3
S1
F0/1
F0/2
F0/3
F0/2
F0/1
172.17.10.27
Non-Designated Ports
Designated Ports
Trunk2
Trunk1
F0/2
F0/1
S2
Designated Ports
Root Ports
F0/11
F0/18
F0/6
PC1
PC2
PC3
172.17.10.21
172.17.10.22
172.17.10.23

Port Roles

- When determining the root port on a switch, the switch compares the path costs on all switch ports
- The switch port with the lowest overall path cost to the root is automatically assigned the root port role because it is closest to the root bridge
- Switch uses the customizable port priority value, or the lowest port ID if both port priority values are the same
- The port ID is the interface ID of the switch port. The port ID is appended to the port priority. For example, switch port F0/1 has a default port priority value of 128.1, where 128 is the configurable port priority value, and .1 is the port ID.
- The port priority values range from 0 - 240, in increments of 16. The default port priority value is 128

Port Roles


Root Port
Designated Port
Trunk3
Root Bridge
F0/1
F0/2
S3
S1
Designated Port
F0/2
F0/1
Designated Port
Trunk2
Trunk1
F0/2
X
F0/2
Non-Designated Port
Root Port
S2
S4
F0/1
F0/2
Trunk4
Root Port
Designated Port

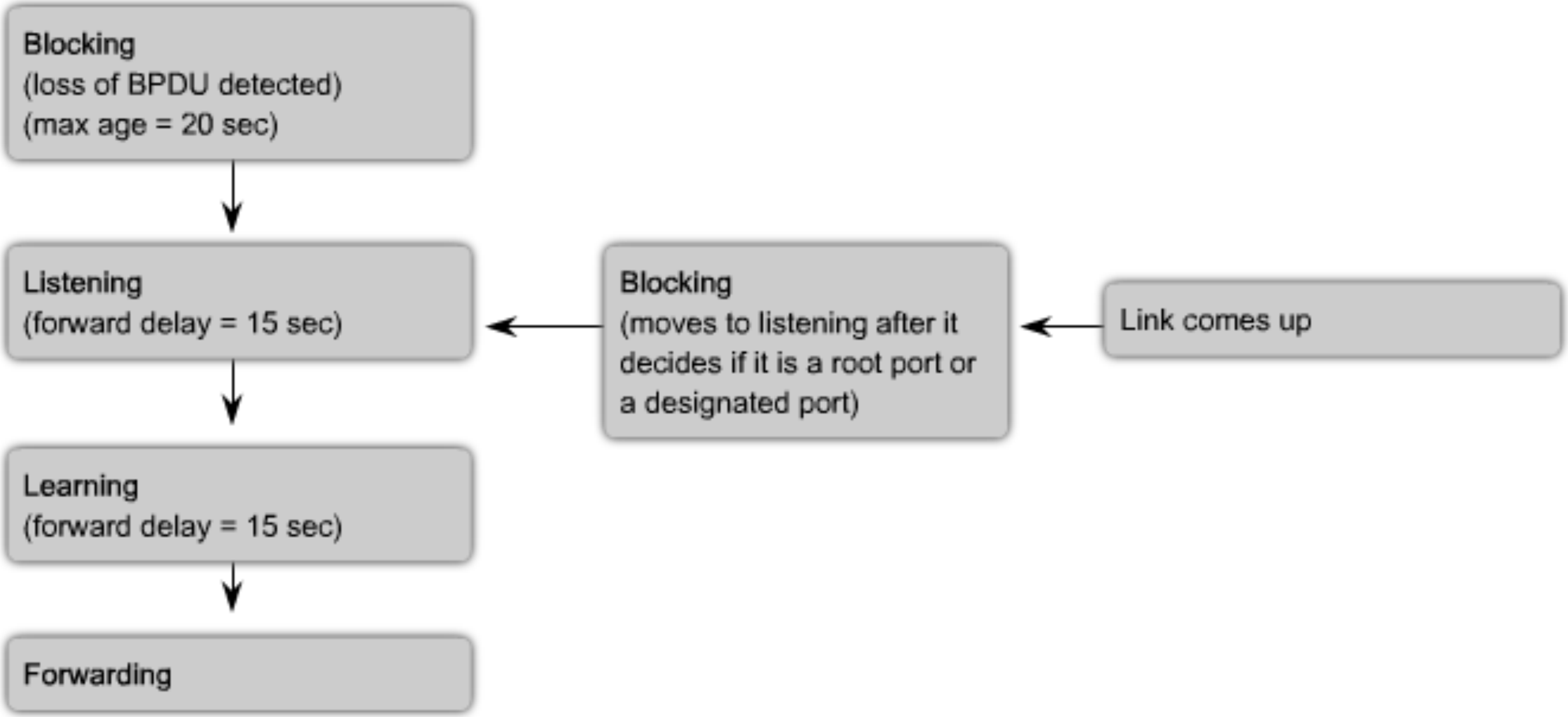
Port States

Processes	Blocking	Listening	Learning	Forwarding	Disable
Receive and process BPDU	√	√	√	√	X
Forward data frames	X	X	X	√	X
Learn MAC addresses	X	X	√	√	X

BPDU Timers

Hello Time	● The hello time is the time between each BPDU frame that is sent on a port. ● This is equal to 2 seconds by default, but can be tuned to be between 1 and 10 seconds.
Forward Delay	● The forward delay is the time spent in the listening and learning state. ● This is by default equal to 15 seconds for each state, but can be tuned to be between 4 and 30 seconds.
Maximum Age	● The max age timer controls the maximum length of time a switch port saves configuration BPDU information. ● This is 20 seconds by default, but can be tuned to be between 6 and 40 seconds.

BPDU Timers



Blocking
(loss of BPDU detected)
(max age = 20 sec)

Listening
(forward delay = 15 sec)

Learning
(forward delay = 15 sec)

Forwarding

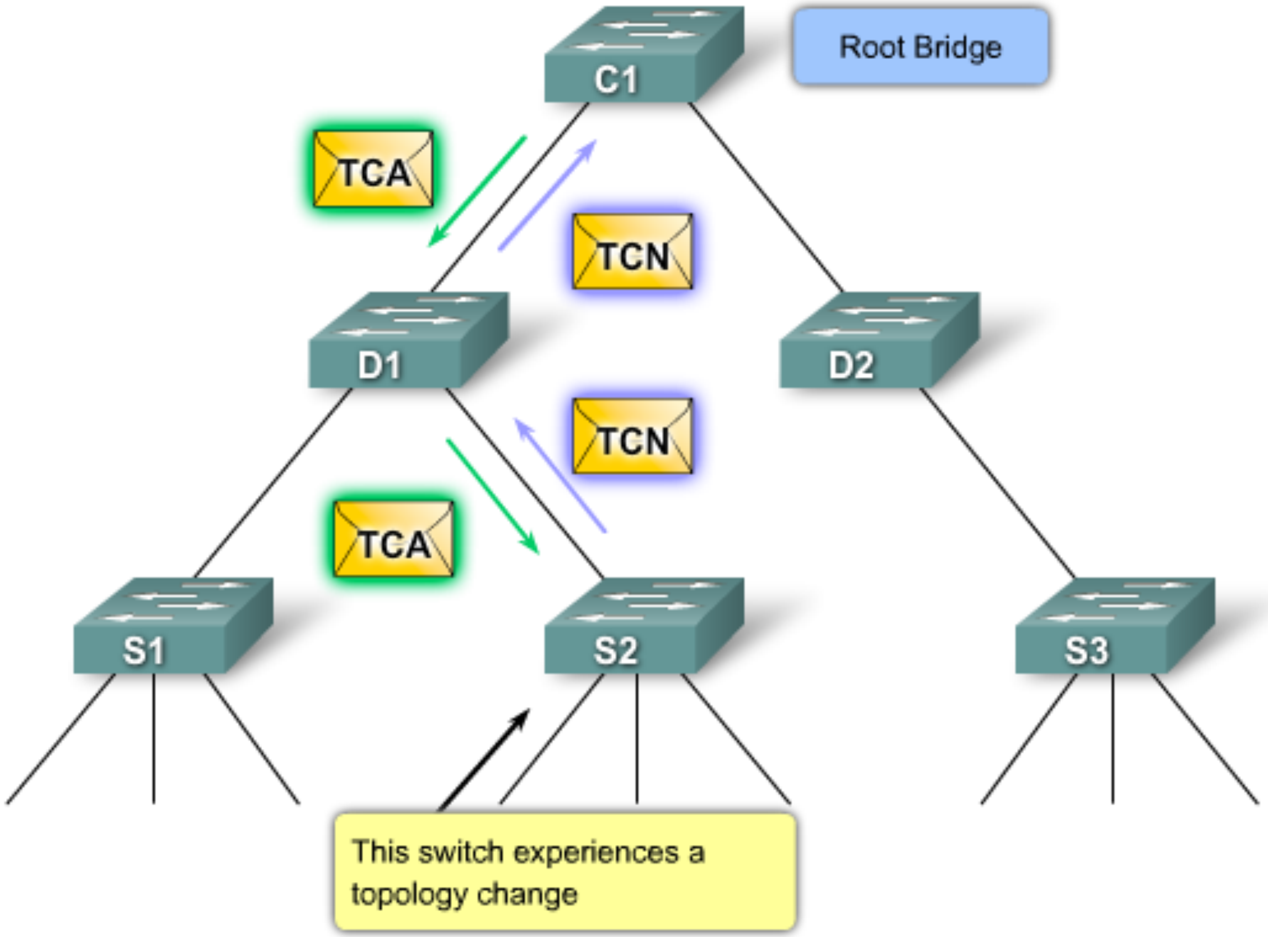
Blocking
(moves to listening after it decides if it is a root port or a designated port)

Link comes up

Root Bridge Election

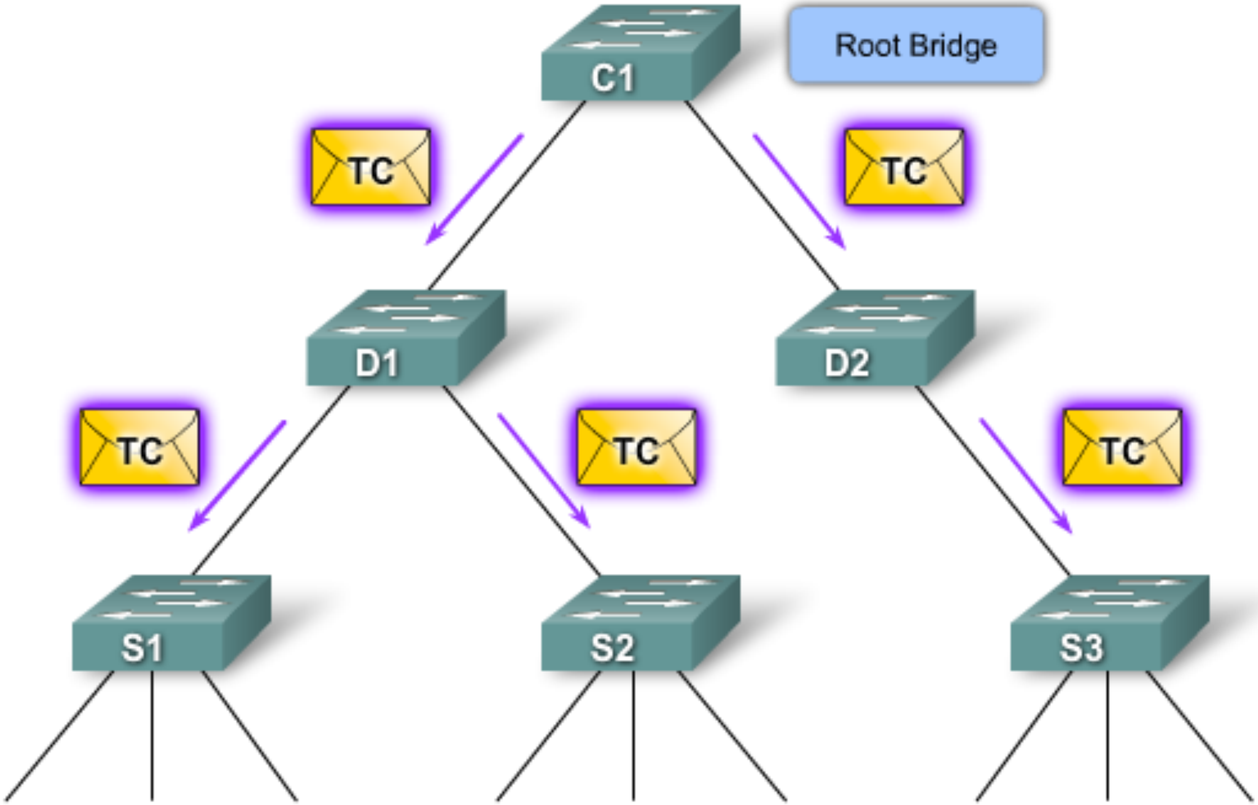
- A root bridge election is triggered after a switch has finished booting up, or when a path failure has been detected on a network.
- Initially, all switch ports are configured for the blocking state, which by default lasts 20 seconds. This is done to prevent a loop from occurring before STP has had time to calculate the best root paths and configure all switch ports to their specific roles.
- While the switch ports are in a blocking state, they are still able to send and receive BPDU frames so that the spanning-tree root election can proceed.
- Spanning tree supports a maximum network diameter of seven switch hops from end to end. This allows the entire root bridge election process to occur within 14 seconds, which is less than the time the switch ports spend in the blocking state.

STP Topology Change

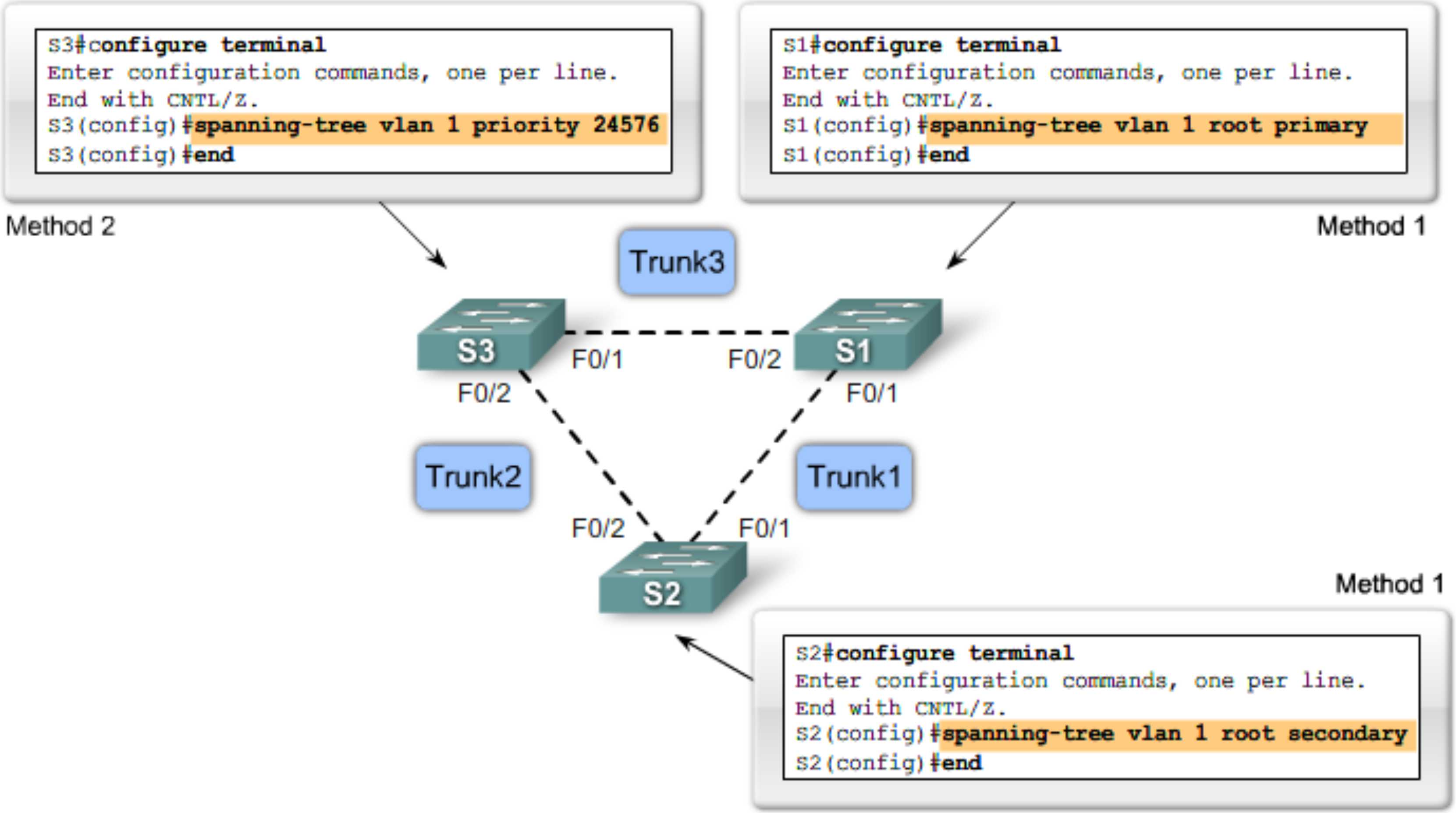


This switch experiences a topology change

STP Topology Change



Configure BID



Method 2

```
S3#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
S3(config)#spanning-tree vlan 1 priority 24576
S3(config)#end
```

Method 1

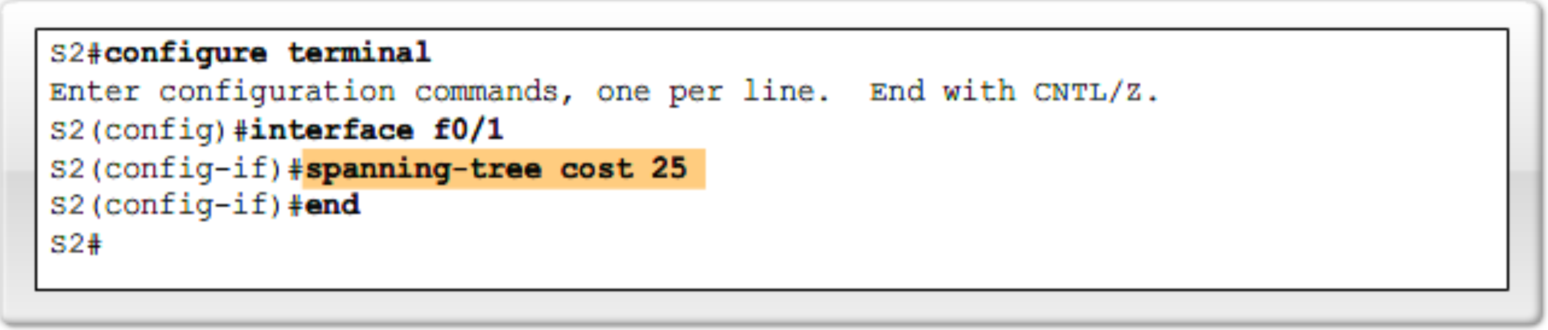
```
S1#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
S1(config)#spanning-tree vlan 1 root primary
S1(config)#end
```

Method 1

```
S2#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
S2(config)#spanning-tree vlan 1 root secondary
S2(config)#end
```

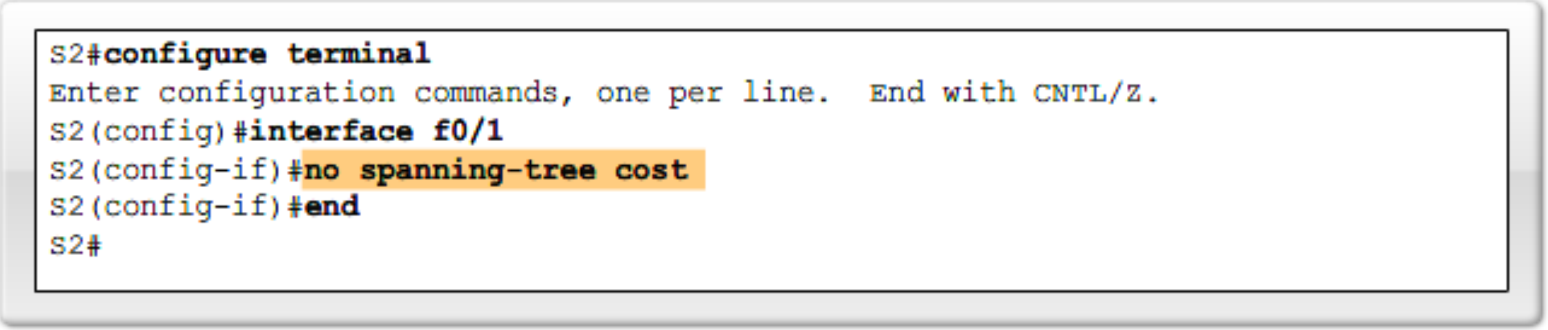
Configure Port Cost

Configure Port Cost



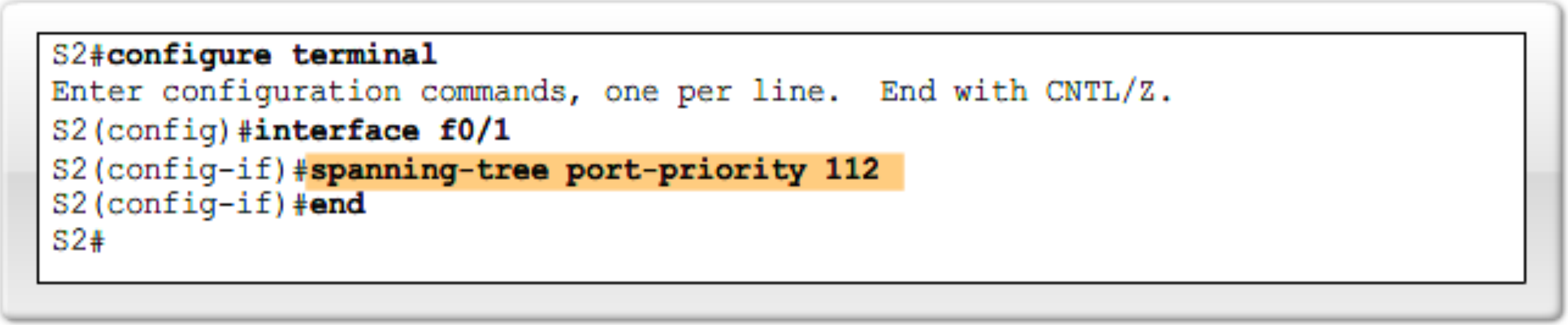
```
S2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
S2(config)#interface f0/1
S2(config-if)#spanning-tree cost 25
S2(config-if)#end
S2#
```

Reset Port Cost



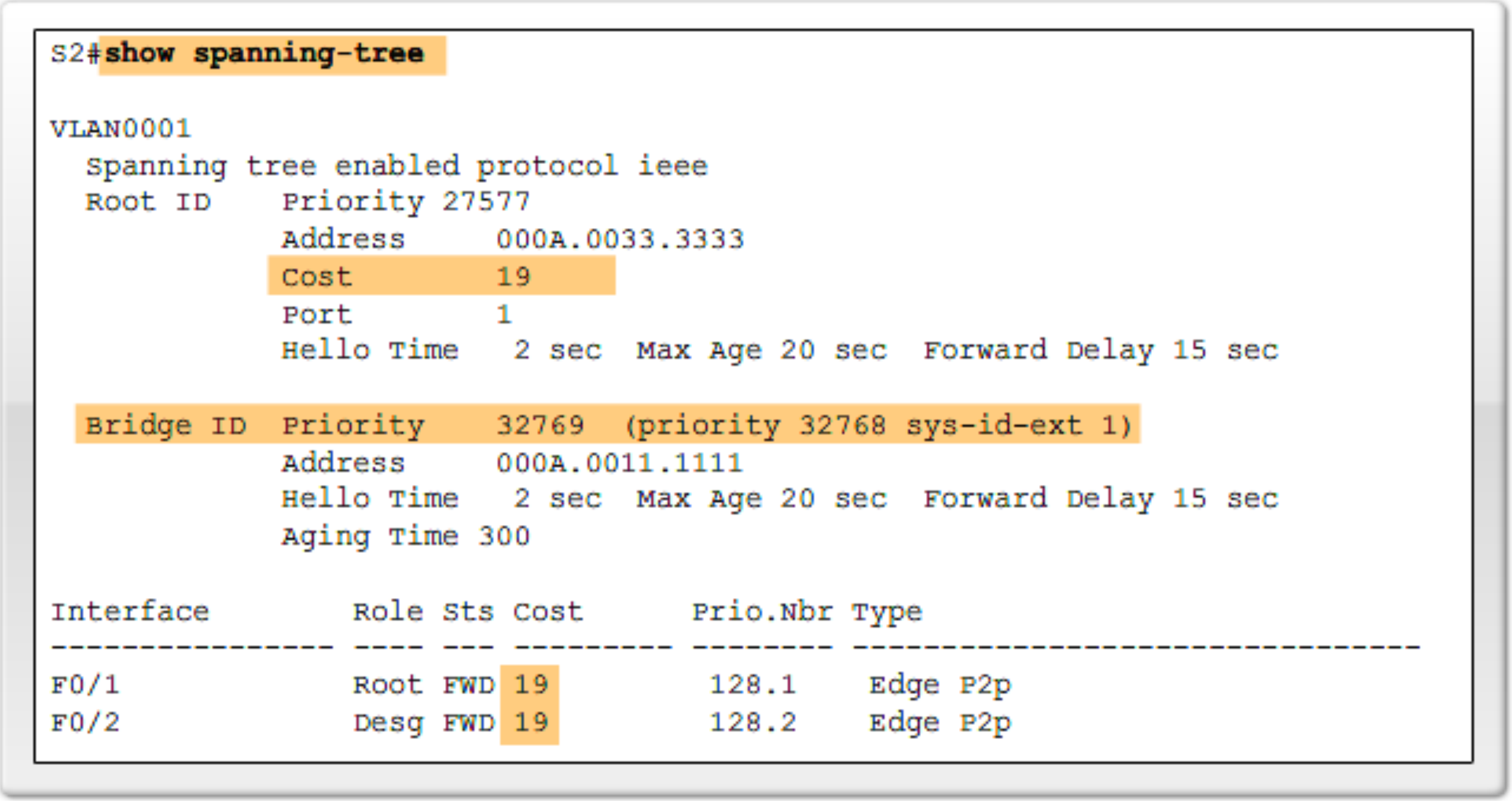
```
S2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
S2(config)#interface f0/1
S2(config-if)#no spanning-tree cost
S2(config-if)#end
S2#
```

Configure Port Priority



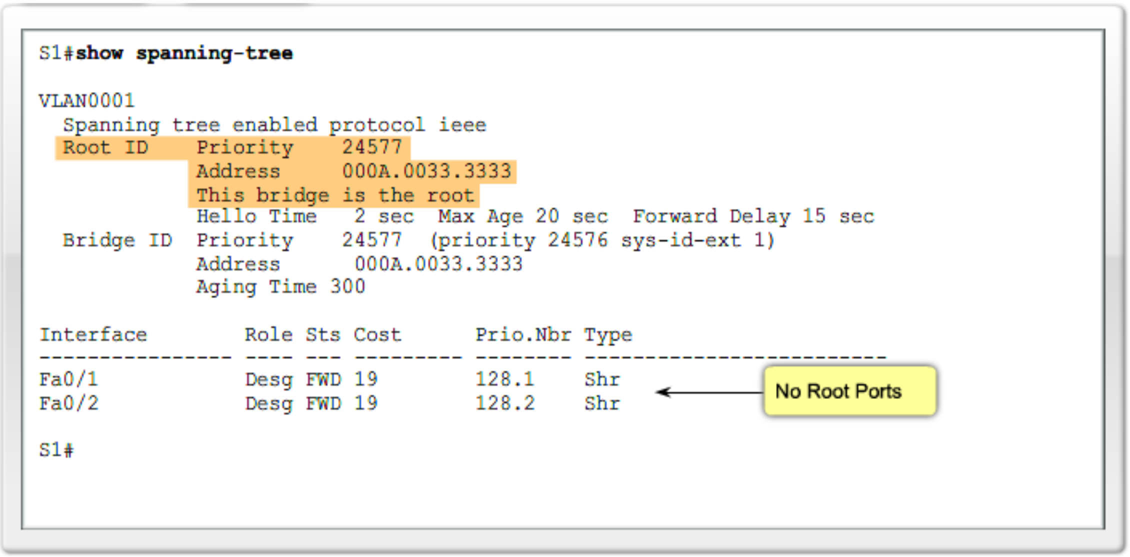
```
S2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
S2(config)#interface f0/1
S2(config-if)#spanning-tree port-priority 112
S2(config-if)#end
S2#
```

Verify BID and Path Cost



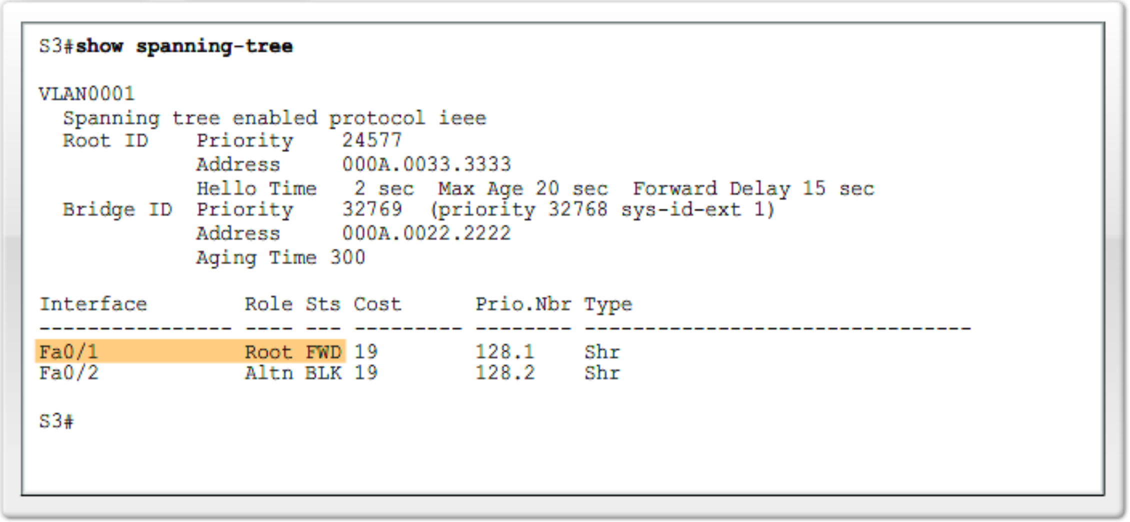
```
S2#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority 27577
              Address      000A.0033.3333
              Cost         19
              Port         1
              Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority     32769  (priority 32768 sys-id-ext 1)
              Address      000A.0011.1111
              Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
F0/1             Root FWD 19         128.1    Edge P2p
F0/2             Desg FWD 19         128.2    Edge P2p
```

Verification



```
S1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     000A.0033.3333
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1)
              Address      000A.0033.3333
              Aging Time 300

Interface         Role Sts Cost      Prio.Nbr Type
----------------- ---- --- --------- -------- -------------------------
Fa0/1             Desg FWD 19        128.1    Shr
Fa0/2             Desg FWD 19        128.2    Shr

S1#
```

No Root Ports

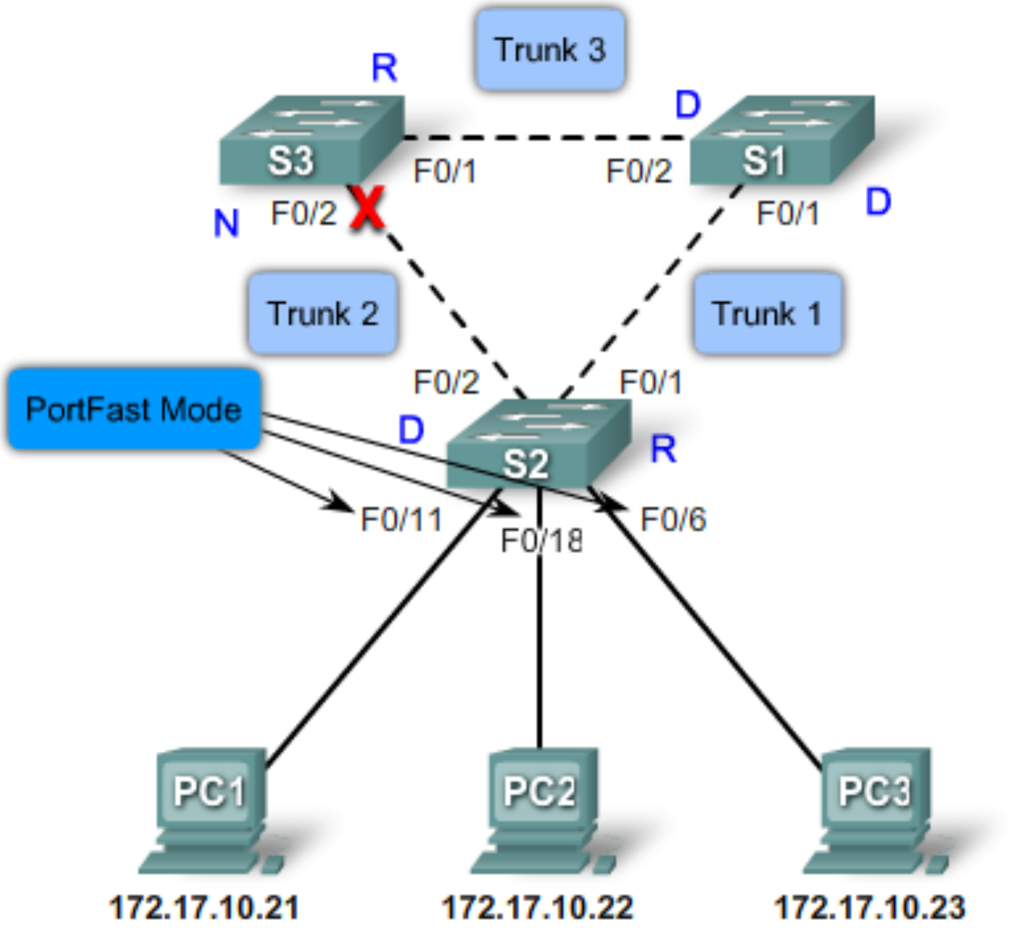


```
S3#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     000A.0033.3333
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     000A.0022.2222
              Aging Time 300

Interface         Role Sts Cost      Prio.Nbr Type
----------------- ---- --- --------- -------- --------------------------------
Fa0/1             Root FWD 19        128.1    Shr
Fa0/2             Altn BLK 19        128.2    Shr

S3#
```

STP PortFast Technology

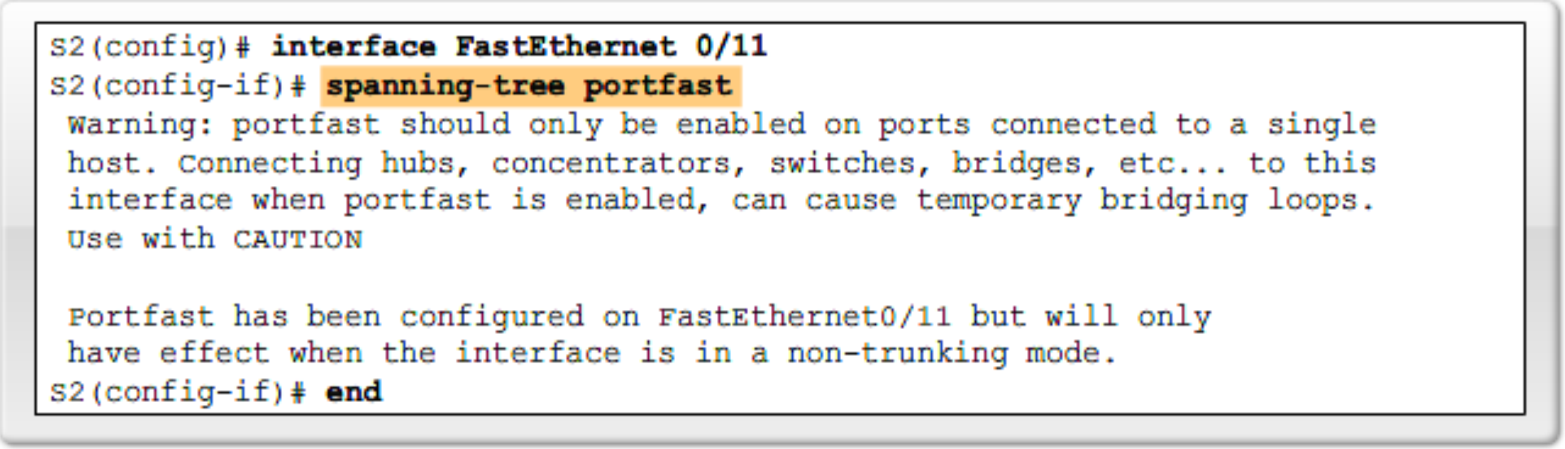
- PortFast is Cisco proprietary
- PortFast configures a switchport as access mode, that port transitions from blocking to forwarding state immediately, bypassing the typical STP listening and learning states
- Because the purpose of PortFast is to minimize the time that access ports must wait for spanning tree to converge, it should be used only on access ports. If you enable PortFast on a port connecting to another switch, you risk creating a spanning-tree loop

STP PortFast Technology



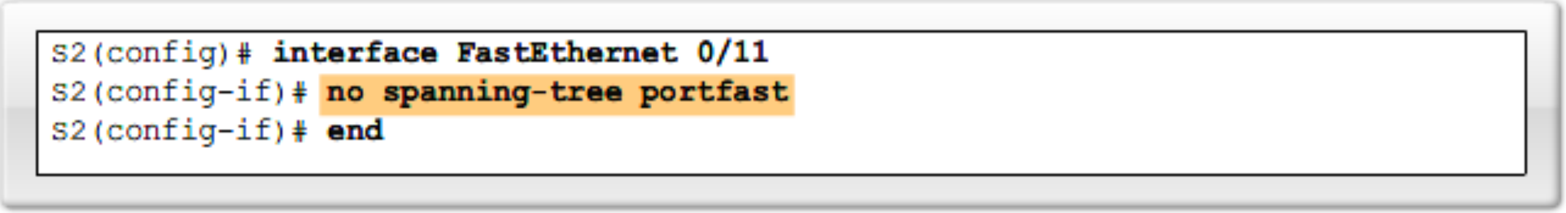
STP PortFast Technology

Enable PortFast



```
S2(config)# interface FastEthernet 0/11
S2(config-if)# spanning-tree portfast
 Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

 Portfast has been configured on FastEthernet0/11 but will only
 have effect when the interface is in a non-trunking mode.
S2(config-if)# end
```

Disable PortFast



```
S2(config)# interface FastEthernet 0/11
S2(config-if)# no spanning-tree portfast
S2(config-if)# end
```

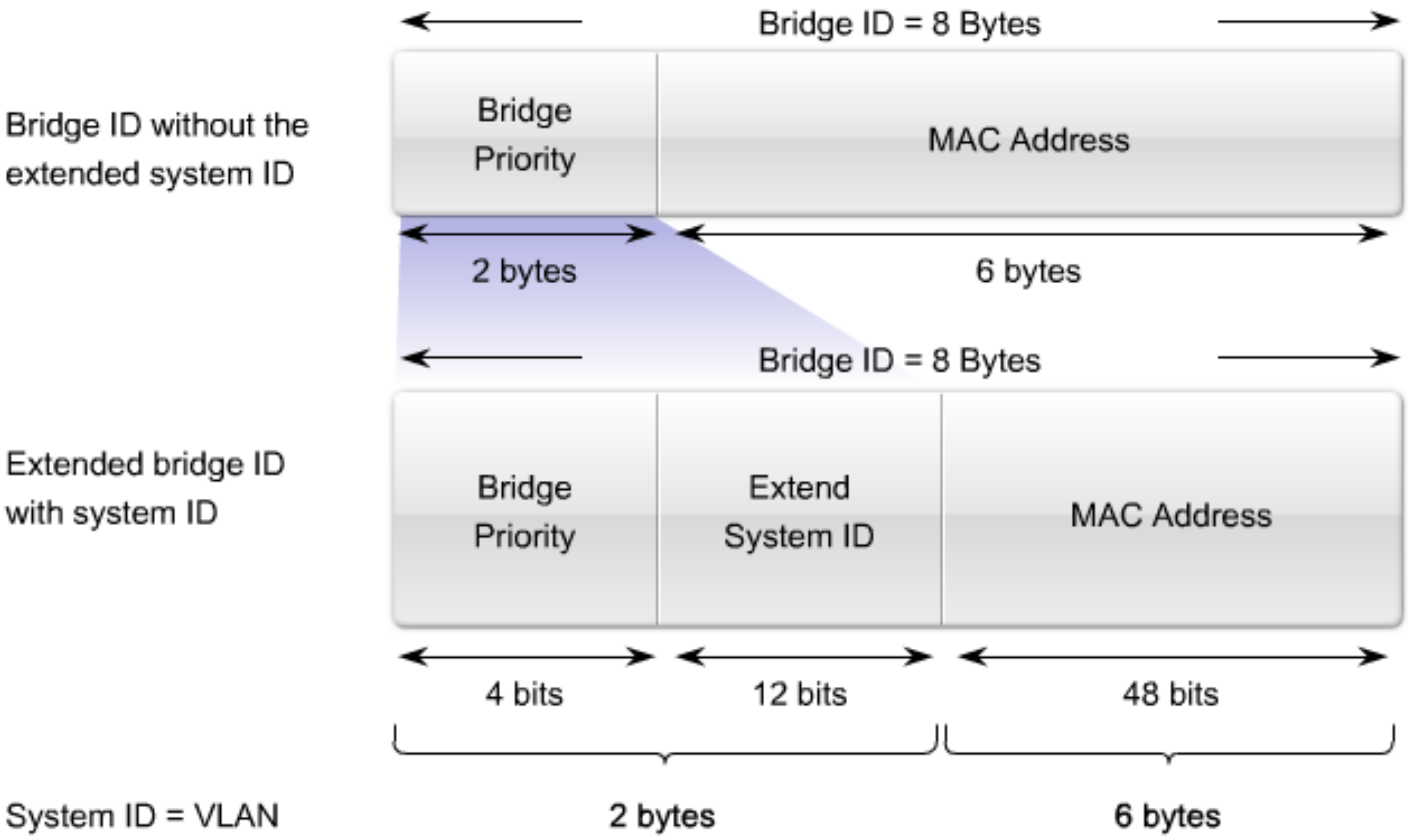
Use the ***spanning-tree portfast default*** global configuration mode command to enable the PortFast feature on all nontrunking interfaces

Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol

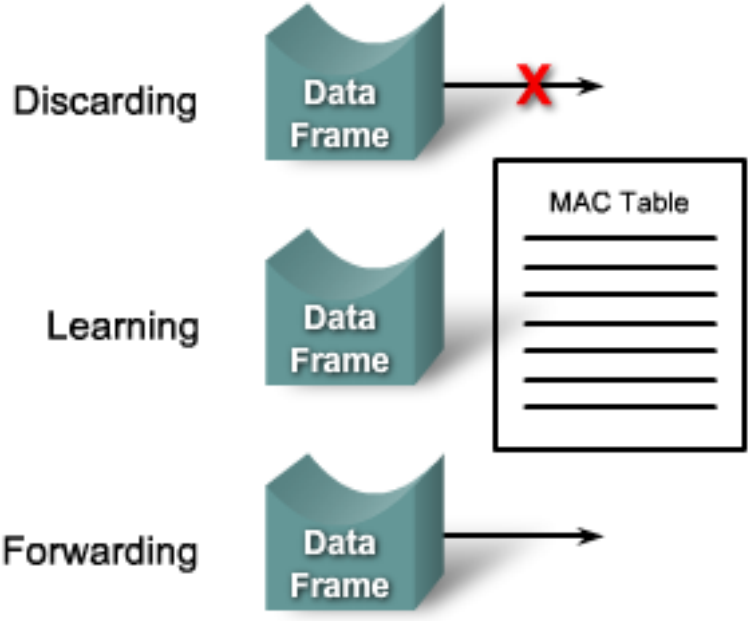
- RSTP (IEEE 802.1w) is backward compatible with STP (IEEE 802.1D) having same BPDU format
- Transparently integrates Cisco proprietary enhancements
- Performs better than Cisco proprietary enhancements
- Not compatible with Cisco proprietary enhancements
- Does not need 802.1D timers

RSTP Bridge ID (BID)



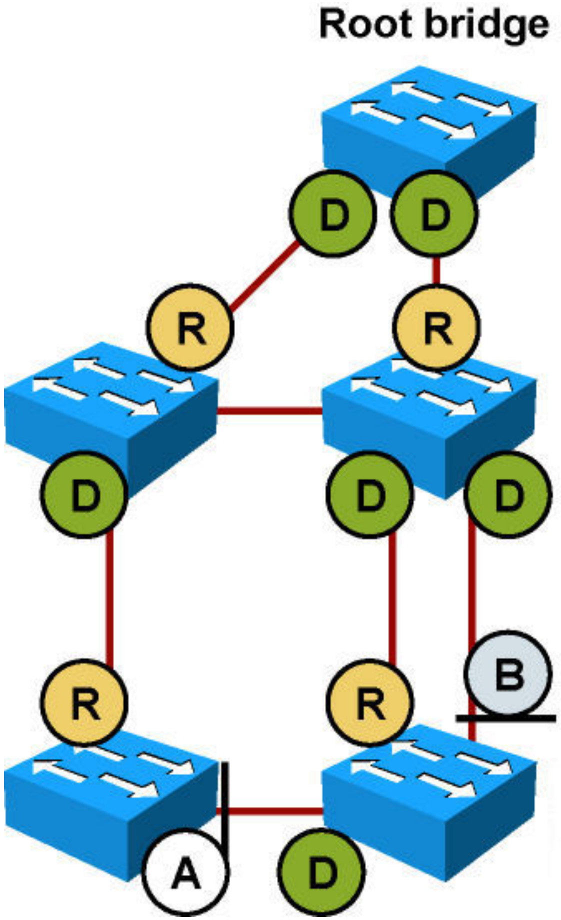
RSTP Port States

STP Port States	RSTP Port States
Disable	Discarding
Blocking	Discarding
Listening	Discarding
Learning	Learning
Forwarding	Forwarding

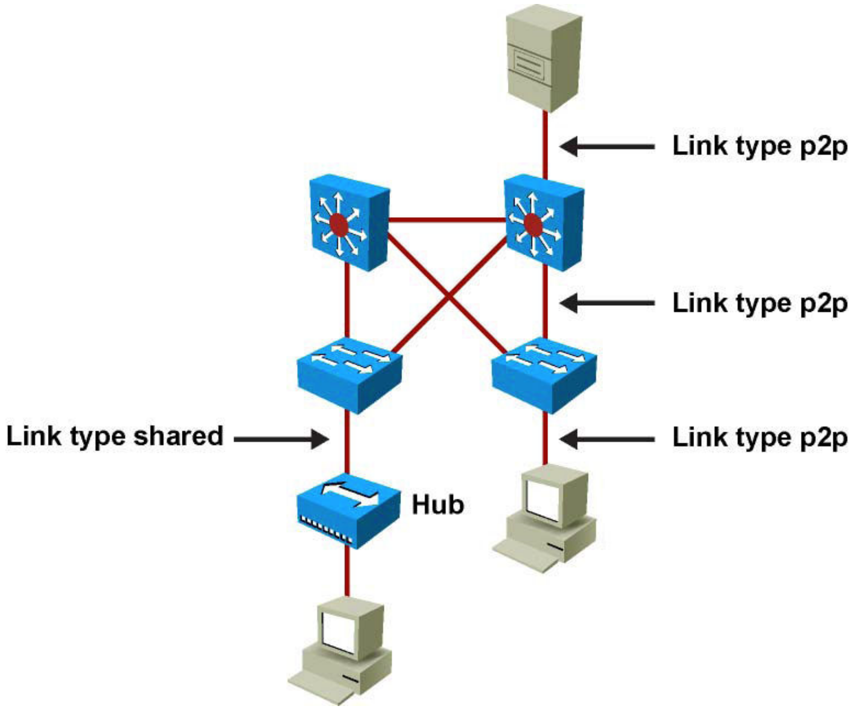


RSTP Port Roles

- Ports in forwarding mode
  - Root (R)
  - Designated (D)
- Ports in blocking mode
  - Alternate (A)
  - Backup (B)

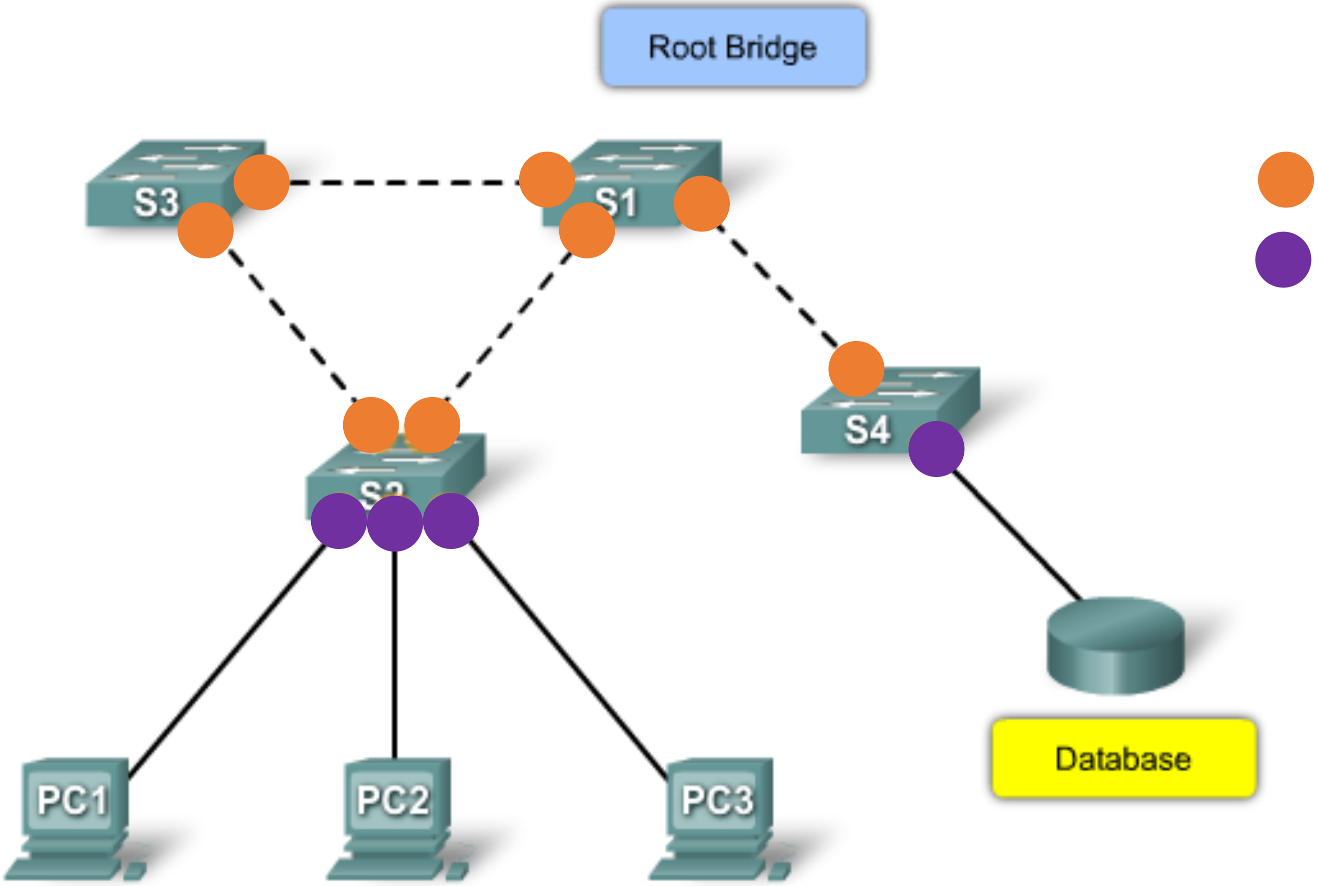


RSTP Link Types



Link Type	Description
Point-to-point (P2p)	Port operating in full-duplex mode. Assumed that the port is connected to a single switch at the other end of the link.
Shared (Shr)	Port operating in half-duplex mode. Assumed that the port is connected to a shared media where multiple switches might exist.

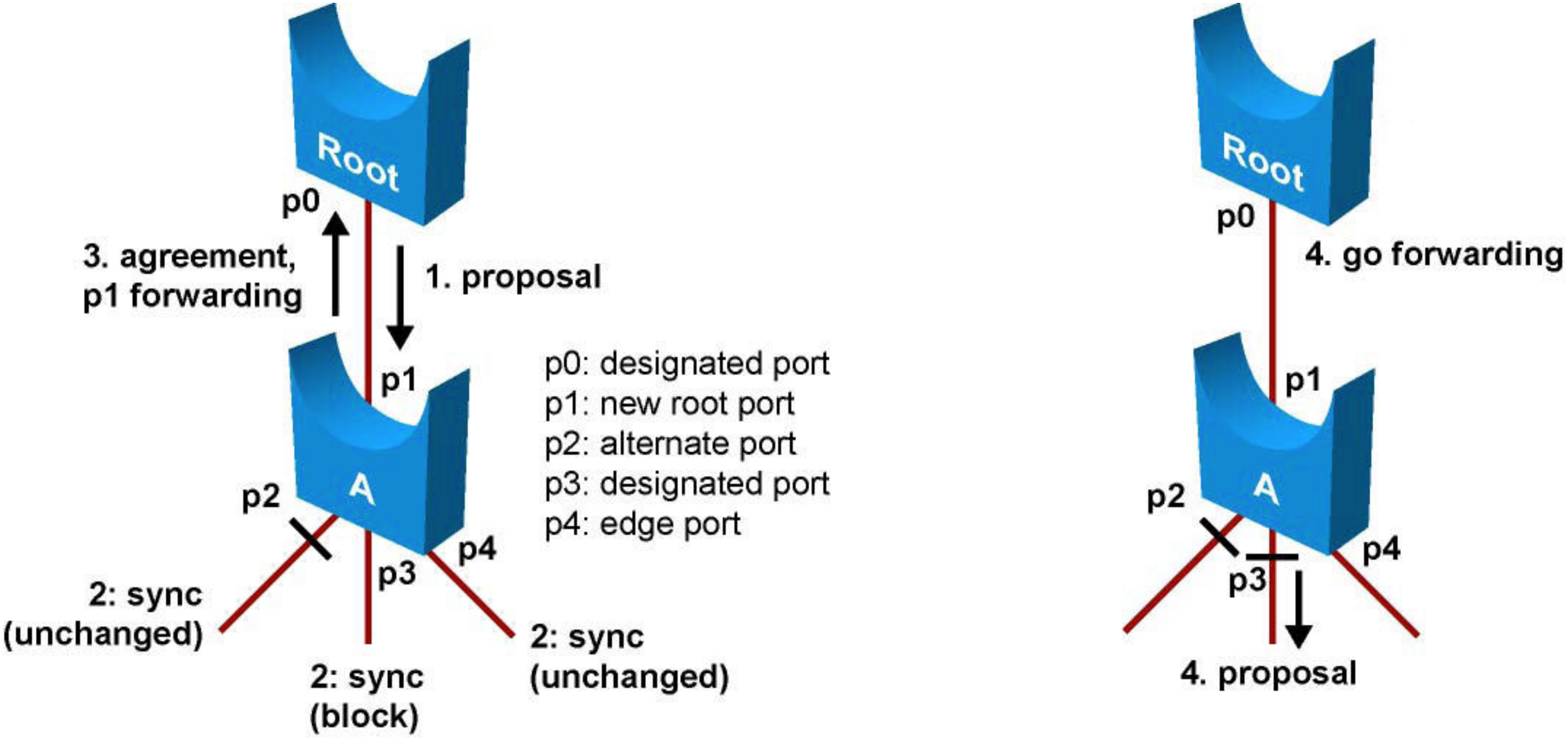
RSTP Port Types



● Non-Edge Ports

● Edge Ports

RSTP Proposal & Agreement Process



p0: designated port
p1: new root port
p2: alternate port
p3: designated port
p4: edge port

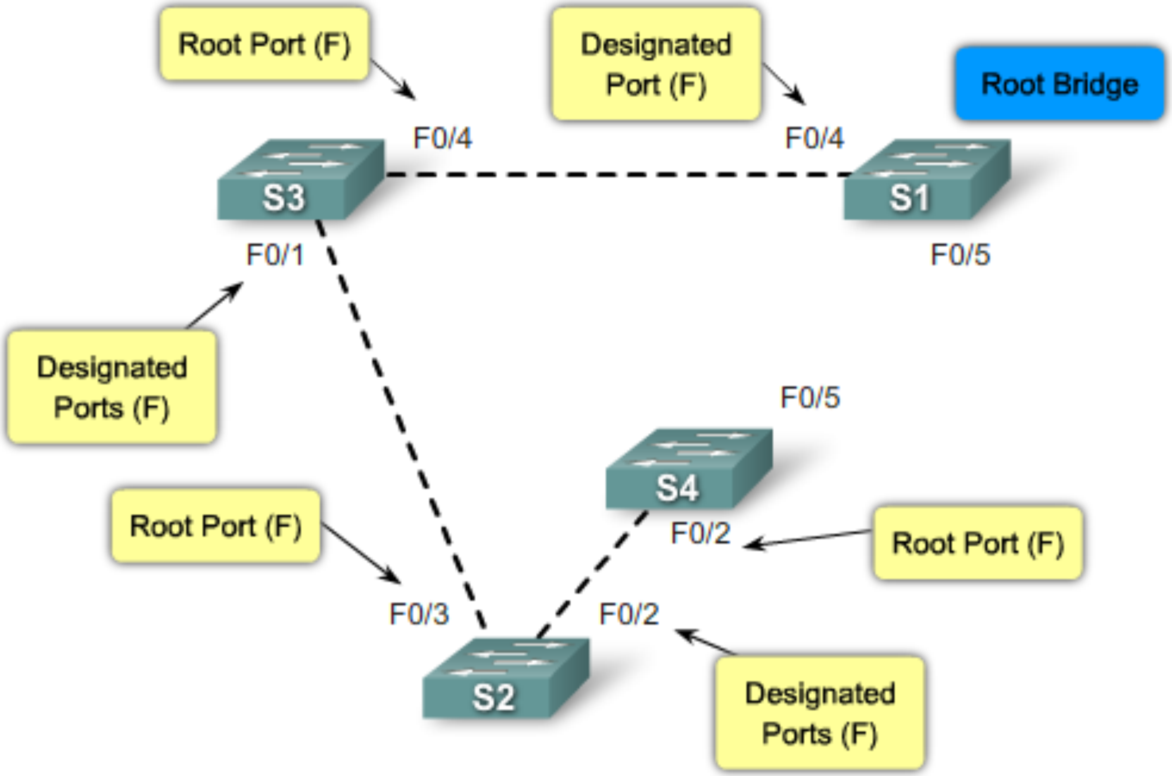
RSTP Proposal & Agreement Process

- A new link is created between the root and Switch A. Both ports are in a designated blocking state
- When new links a designated port is in a discarding or learning state, it sets the proposal bit on the BPDUs it sends out. This is what occurs for port p0 of the root bridge, as shown in Step 1. Because Switch A receives superior information, it immediately knows that p1 is the new root port.
- Switch A starts a sync process that puts nonedge designated ports (p3) in blocking state.
- Once all ports are sync, Switch A can unblock its newly selected root, Port p1, and send an agreement message to reply to the root. This message is a copy of the proposal BPDU with the agreement bit set instead of the proposal bit. This ensures that Port p0 knows exactly to which proposal the agreement it receives corresponds.

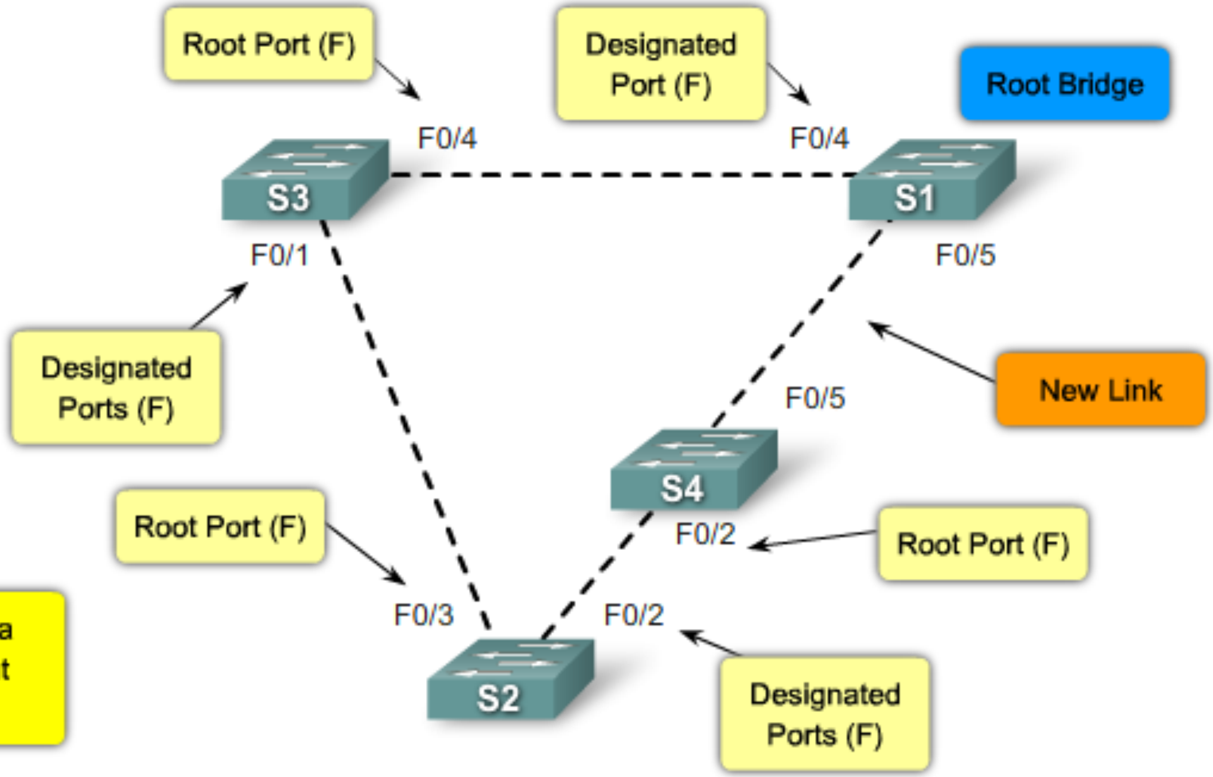
RSTP Proposal & Agreement Process

- When p0 receives that agreement, it can immediately transition to the forwarding state. Root then starts to propose to its neighbor and attempts to quickly transition to the forwarding state.
- If a designated discarding port does not receive an agreement after it sends a proposal, it slowly transitions to the forwarding state by falling back to the traditional 802.1D listening-learning sequence.
- When a bridge loses its root port, it can put its best alternate port directly into forwarding mode. The selection of an alternate port as the new root port generates a topology change.

RSTP Operation

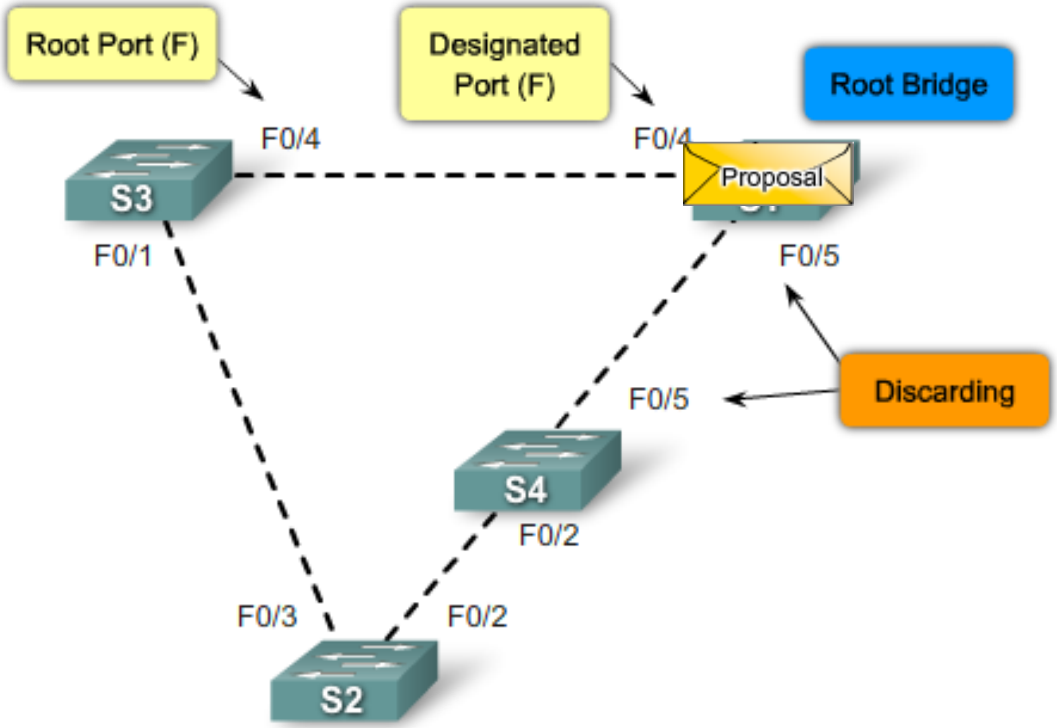


RSTP Operation



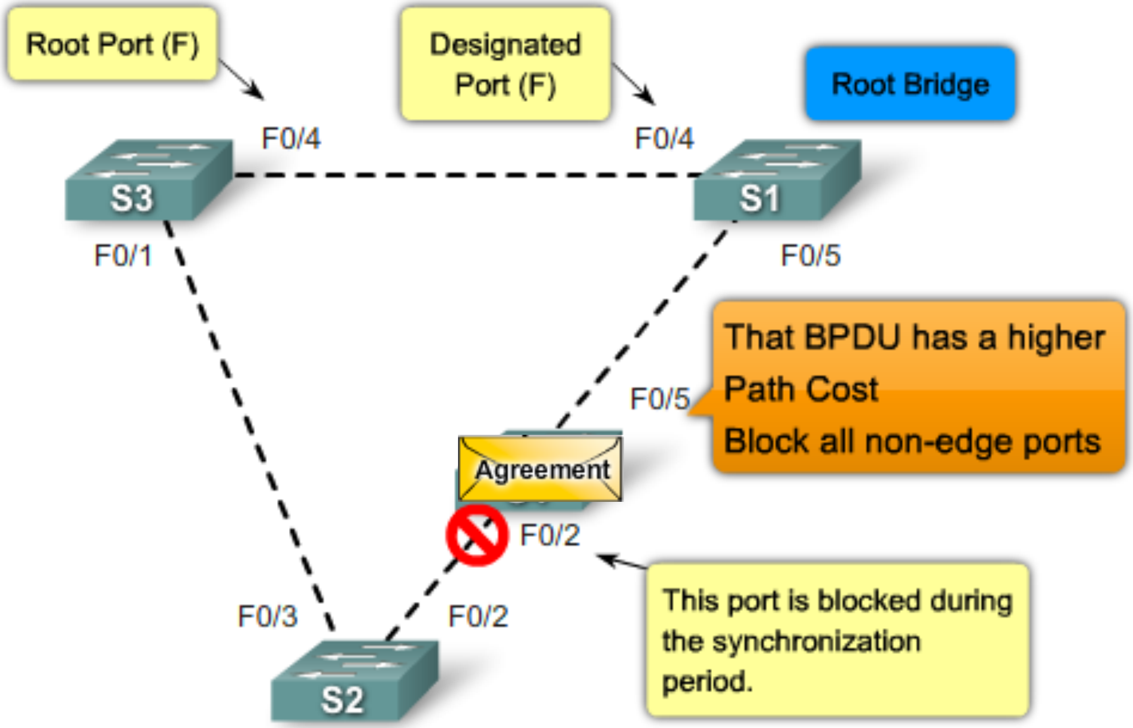
Switch S1 and S4 start a proposal and agreement process

RSTP Operation



Switch S1 sends S4 a proposal BPDU

RSTP Operation



Root Port (F)

Designated Port (F)

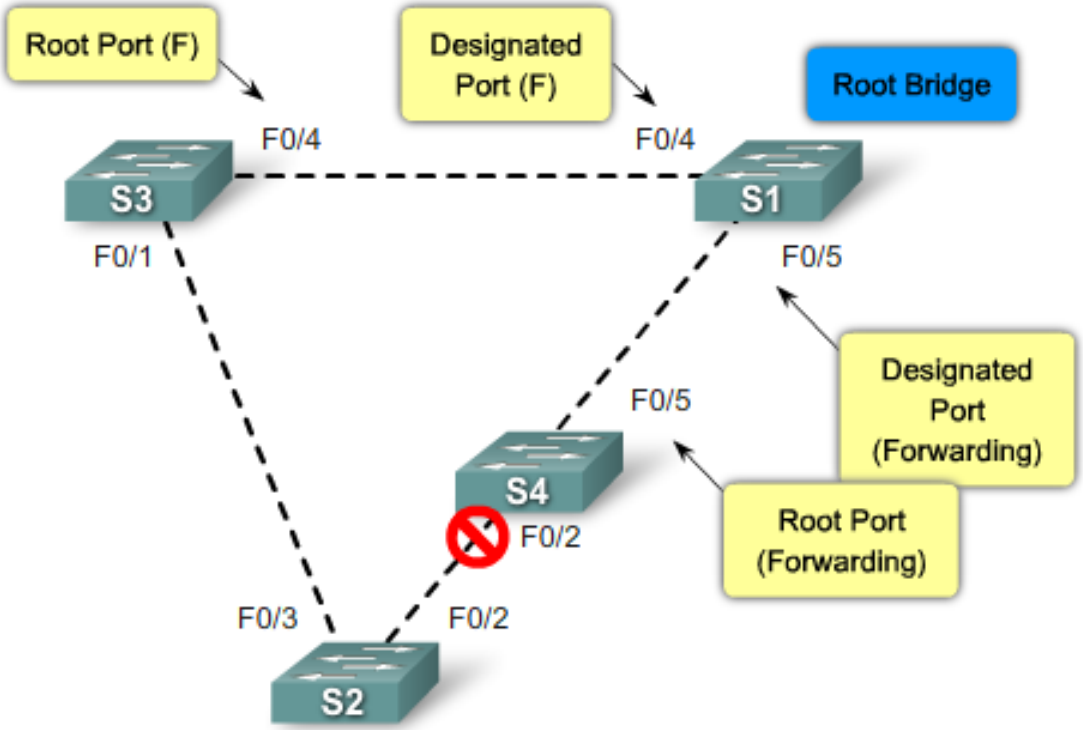
That BPDU has a higher Path Cost

Block all non-edge ports

This port is blocked during the synchronization period.

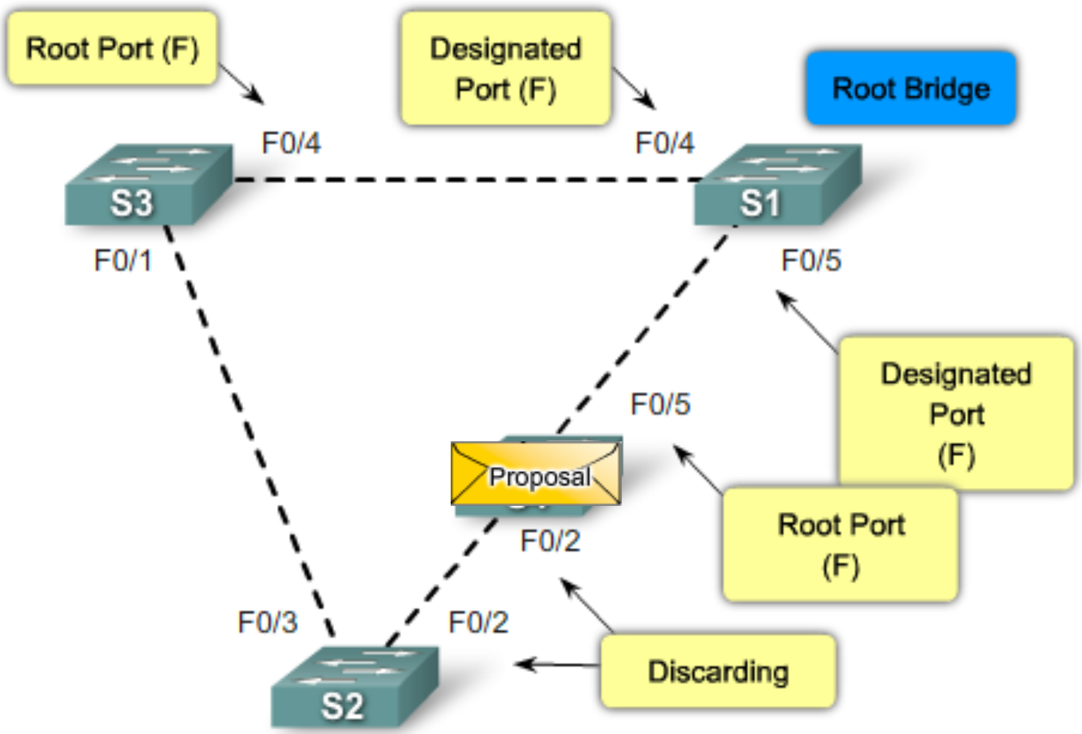
Switch S4 sends S1 an agreement

RSTP Operation



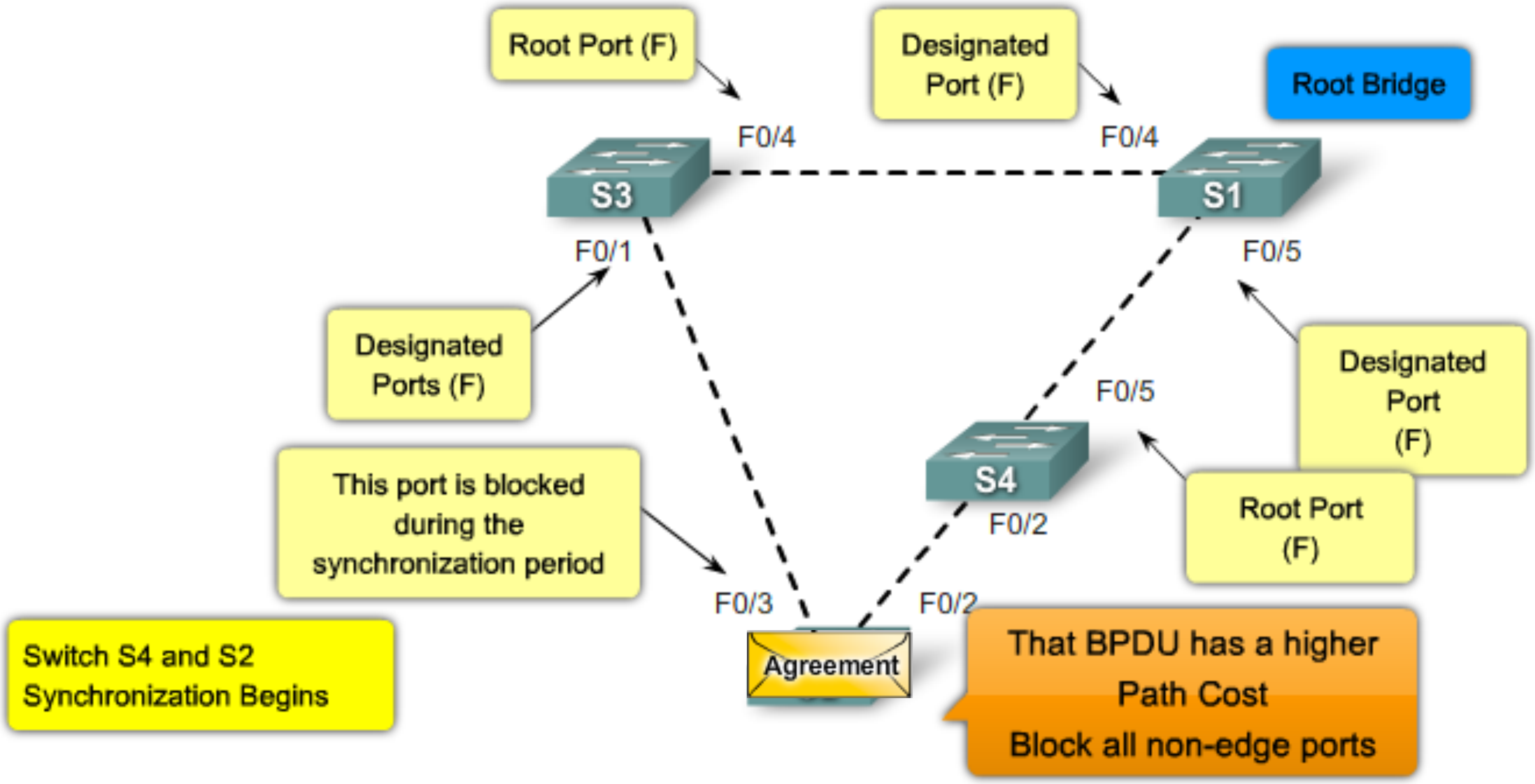
Switch S1 and S4 Synchronization Ends

RSTP Operation

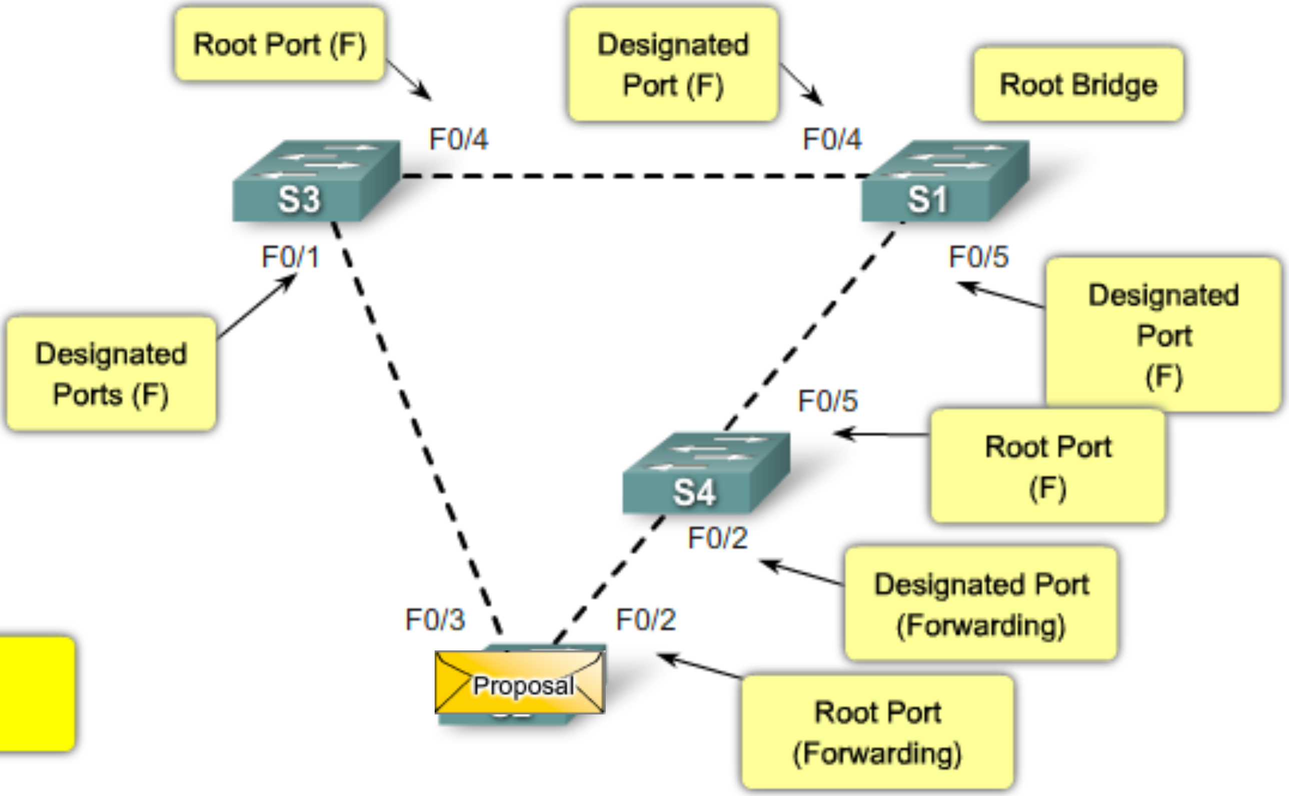


Switch S4 sends S2 a proposal BPDU

RSTP Operation



RSTP Operation



Root Port (F)

Designated Port (F)

Root Bridge

Designated Ports (F)

Designated Port (F)

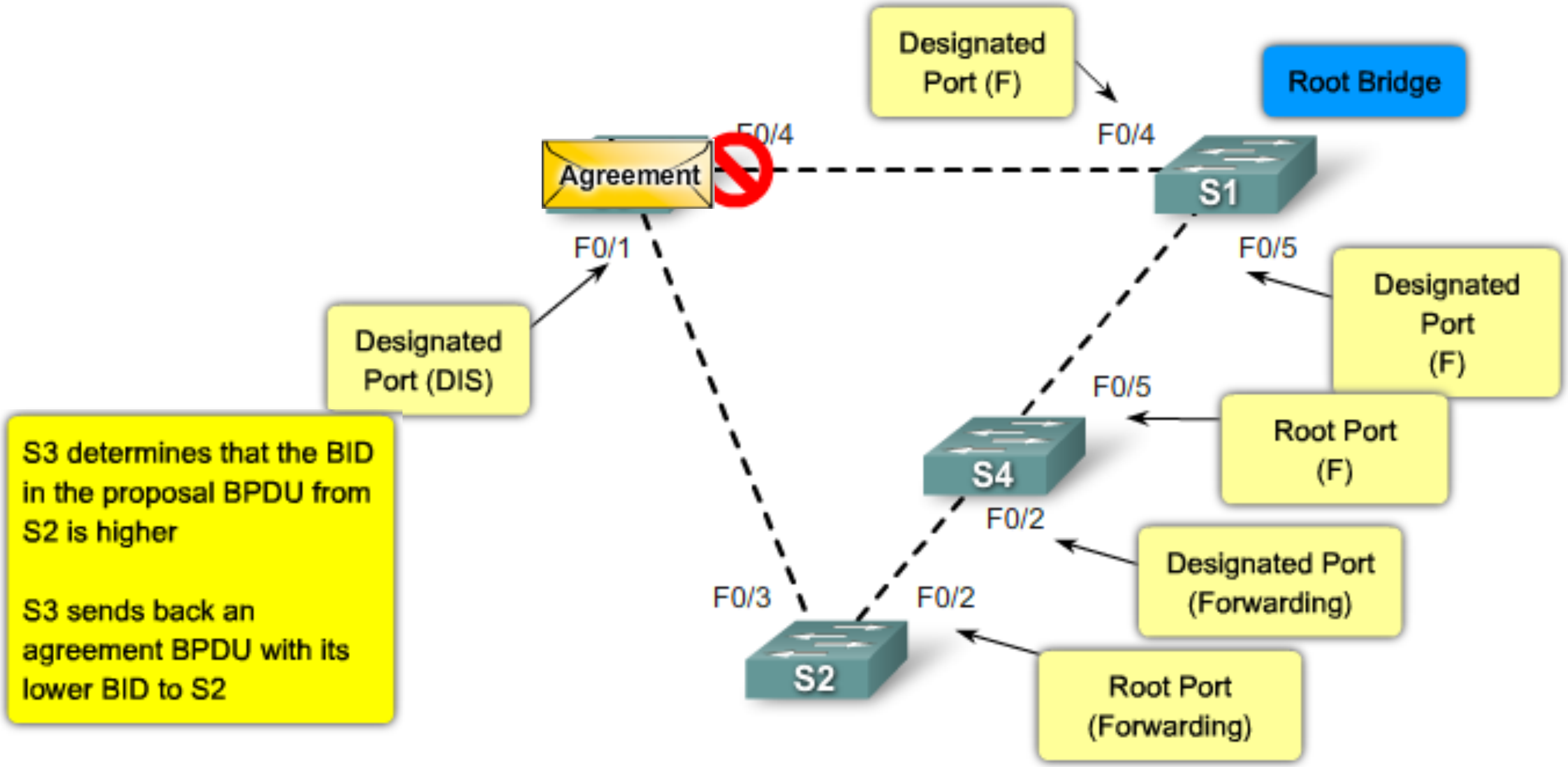
Root Port (F)

Designated Port (Forwarding)

Root Port (Forwarding)

BPDU is exchanged between s2 and s3

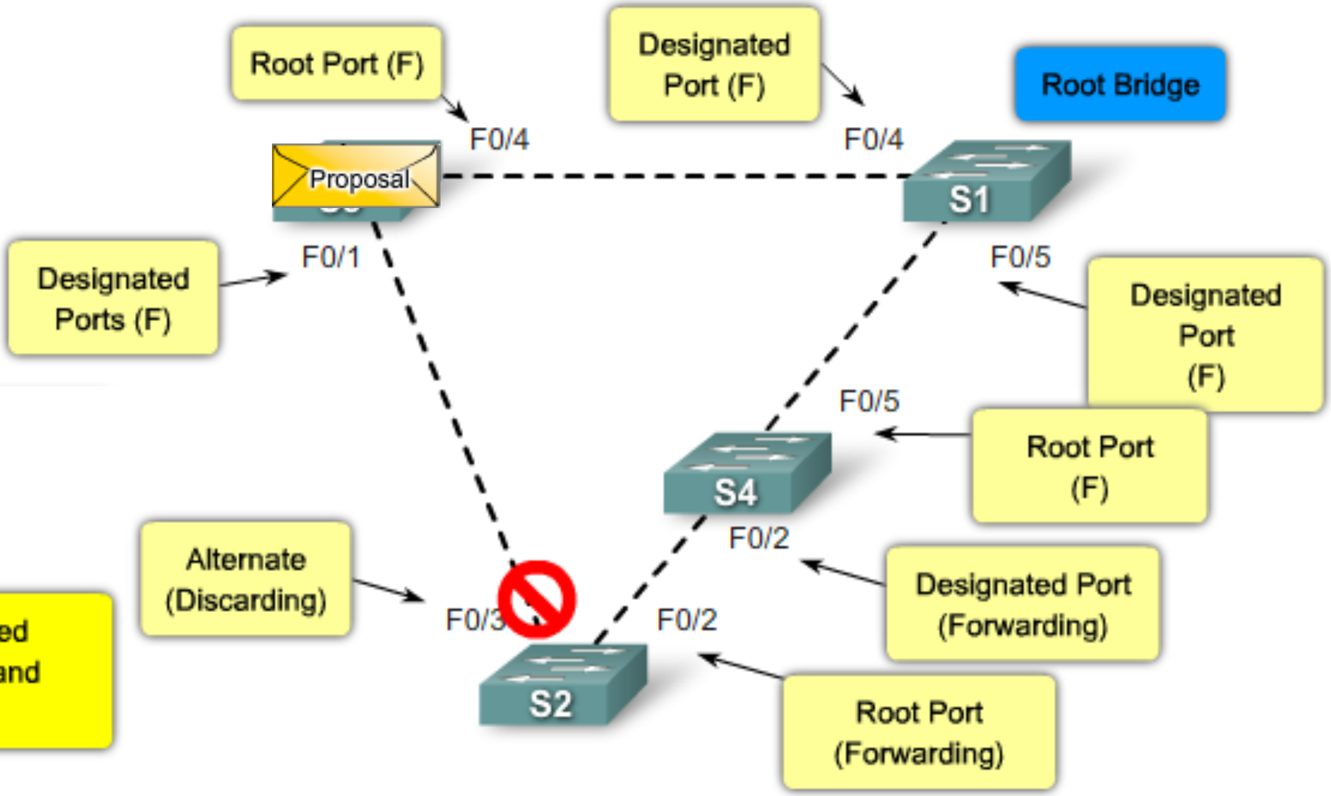
RSTP Operation



S3 determines that the BID in the proposal BPDU from S2 is higher

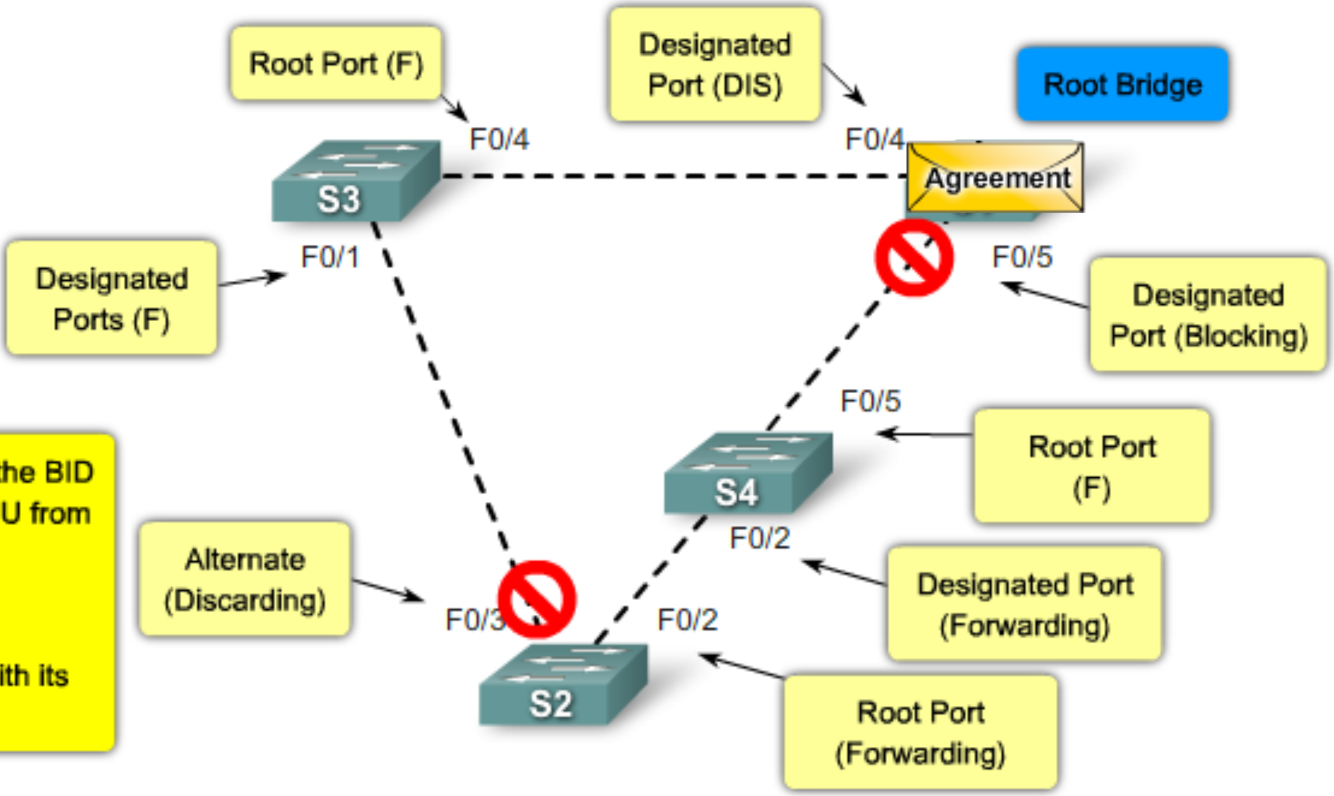
S3 sends back an agreement BPDU with its lower BID to S2

RSTP Operation



A BPDU is exchanged between switch S3 and switch S1

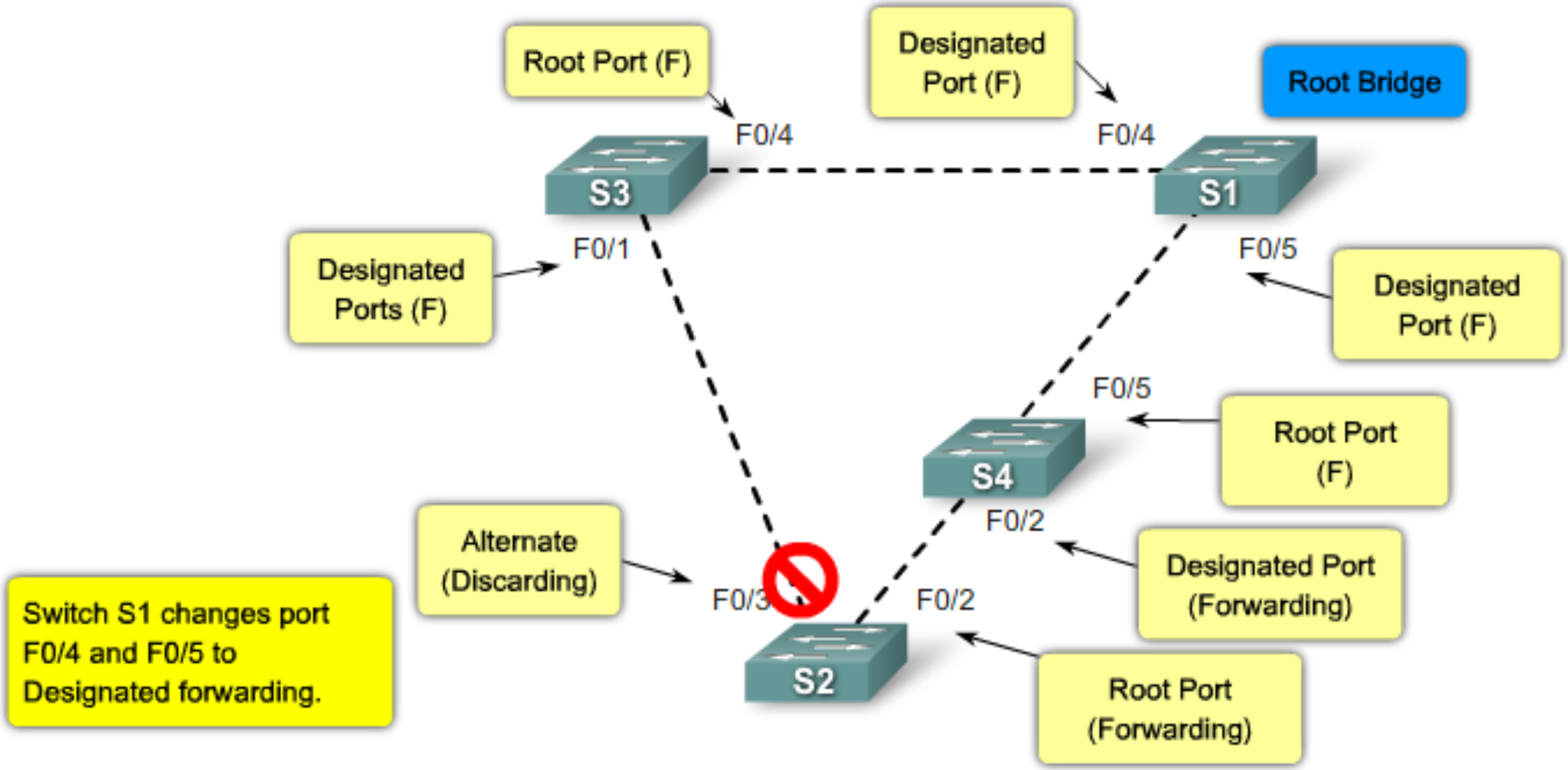
RSTP Operation



S1 determines that the BID in the proposal BPDU from S3 is higher

S1 sends back an agreement BPDU with its lower BID to S3

RSTP Operation

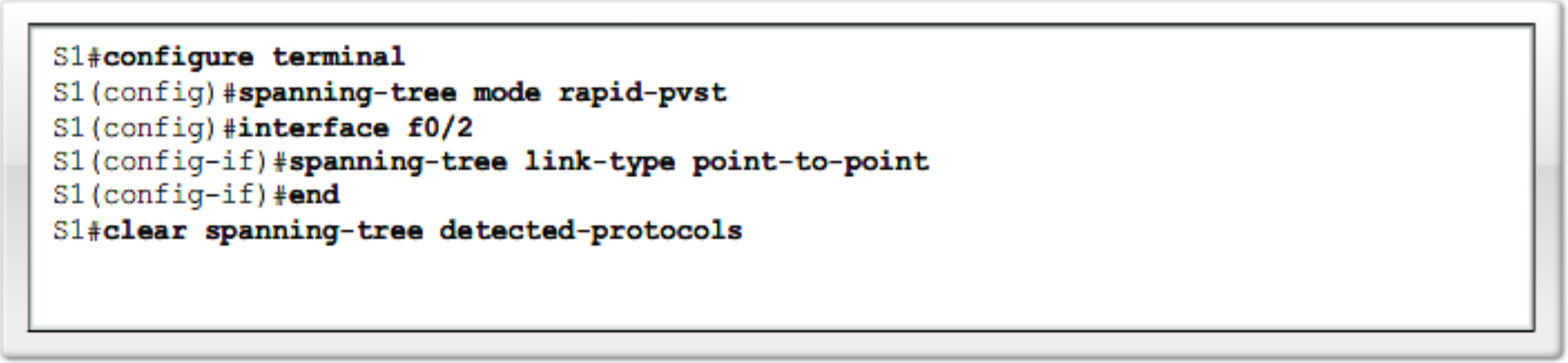


Switch S1 changes port F0/4 and F0/5 to Designated forwarding.

RSTP Topology Change (TC)

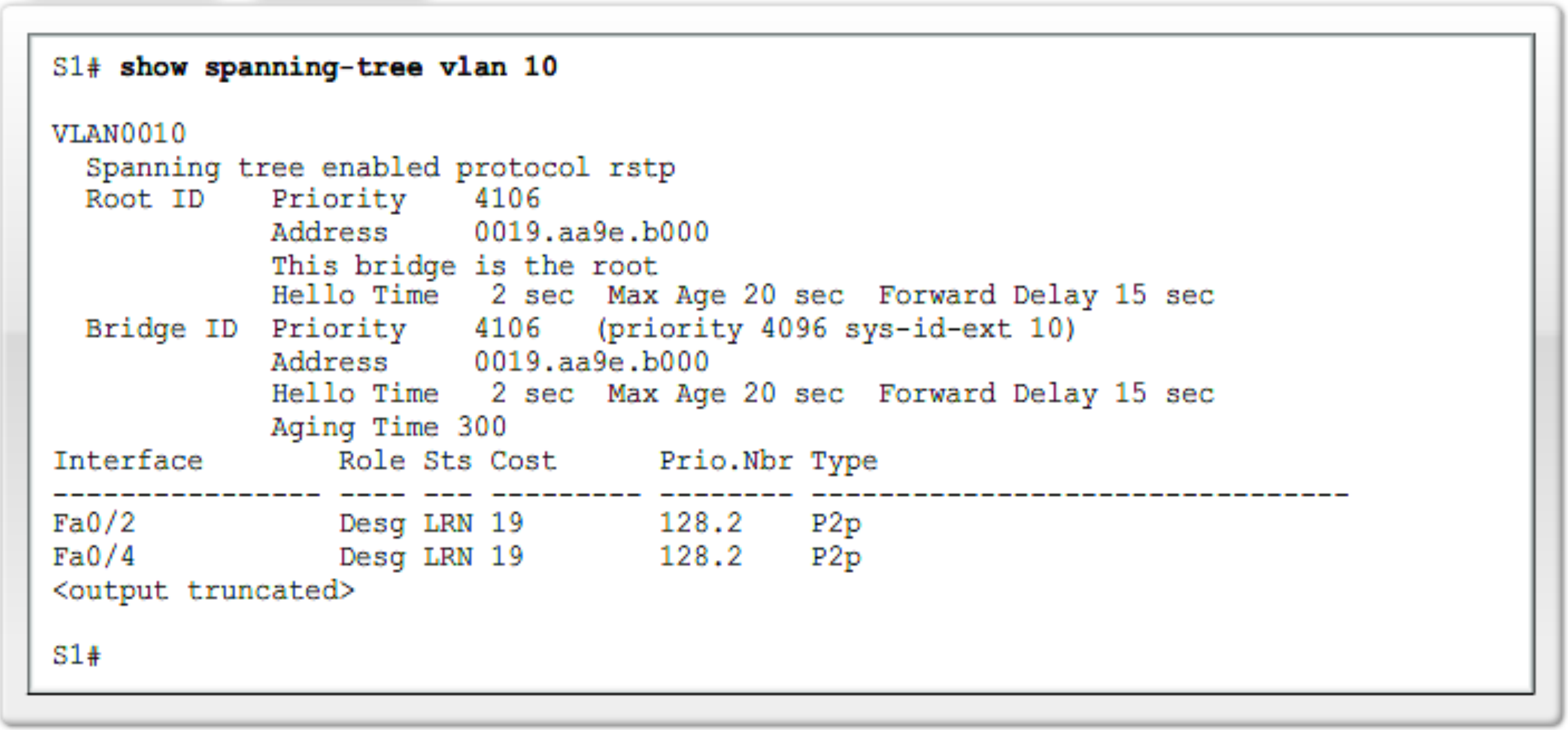
- The RSTP bridge starts the TC While timer with a value equal to twice the hello time for all its nonedge designated ports and its root port, if necessary. The TC While timer is the interval during which the RSTP bridge actively informs the rest of the bridges in the network of a topology change.
- The RSTP bridge flushes the MAC addresses associated with all nonedge ports.
- As long as the TC While timer is running on a port, the BPDUs sent out of that port have the TC bit set. While the timer is active, the bridge sends BPDUs even on the root port.
- When a bridge receives a BPDU with the TC bit set from a neighbor, the bridge performs these actions:
  - The bridge clears the MAC addresses learned on all its ports, except the one that received the topology change.
  - The bridge starts the TC While timer and sends BPDUs with TC set on all its designated ports and root port; RSTP does not use the specific TCN BPDU anymore unless a legacy bridge needs to be notified.

Rapid-PVST



```
S1#configure terminal
S1(config)#spanning-tree mode rapid-pvst
S1(config)#interface f0/2
S1(config-if)#spanning-tree link-type point-to-point
S1(config-if)#end
S1#clear spanning-tree detected-protocols
```

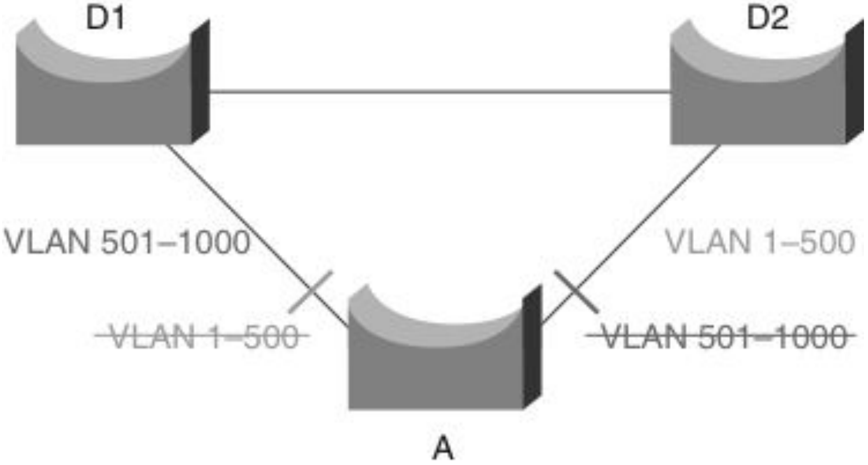
Rapid-PVST Verification



```
S1# show spanning-tree vlan 10

VLAN0010
  Spanning tree enabled protocol rstp
  Root ID     Priority    4106
              Address     0019.aa9e.b000
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID   Priority    4106   (priority 4096 sys-id-ext 10)
              Address     0019.aa9e.b000
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ------------------------------
Fa0/2            Desg LRN 19        128.2    P2p
Fa0/4            Desg LRN 19        128.2    P2p
<output truncated>

S1#
```

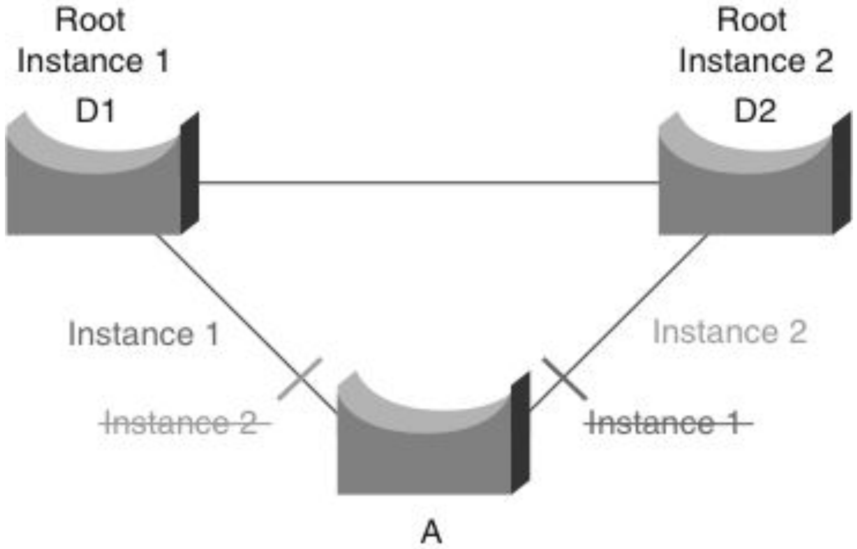
Multiple Spanning Tree

MST Motivation



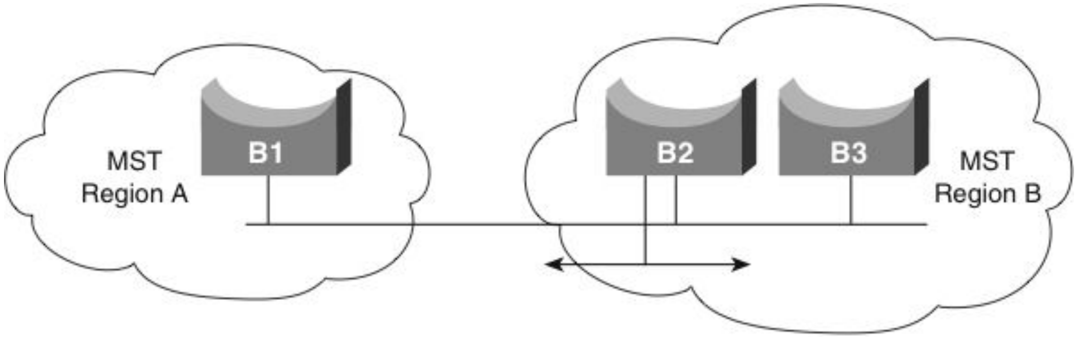
- Above: 2 links – 1000 VLANs – 2 MST instances.
- Each switch maintains only two spanning trees, reducing the need for switch resources.
- Concept extendable to 4096 VLANs: VLAN load balancing.
- MST converges faster than PVRST+ and is backward compatible with 802.1D STP and 802.1w.

MST Instances



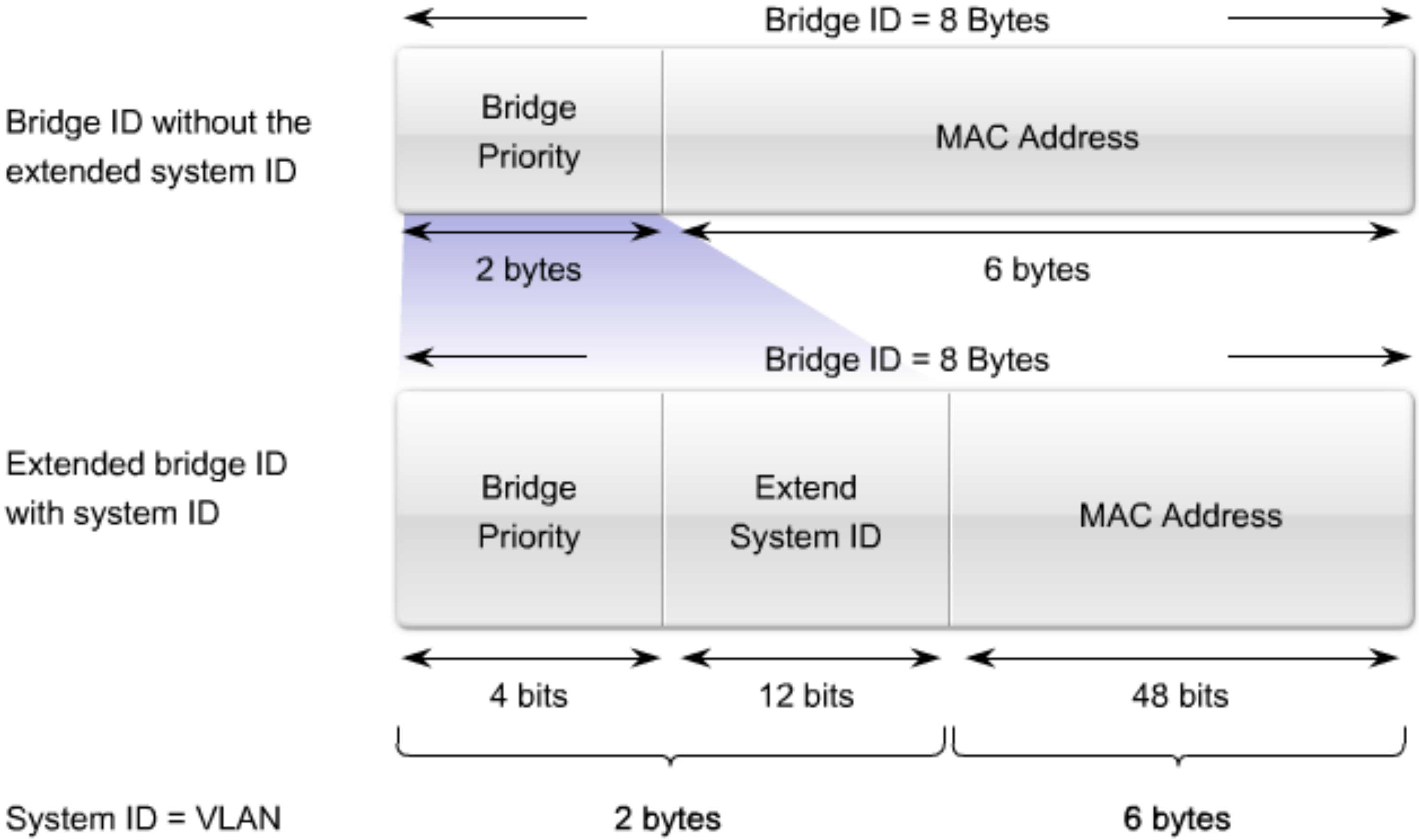
- 2 distinct STP topologies require 2 MST instances (500 per instance here).
- Load-balancing works because half of the VLANs follow each separate instance.
- Switch utilization is low because it only has to handle two instances.
- MST is the best solution for this scenario.
- Considerations: MST is more complex than 802.1D and 802.1w, so it requires additional training. Interaction with legacy bridges can be challenging.

MST Regions

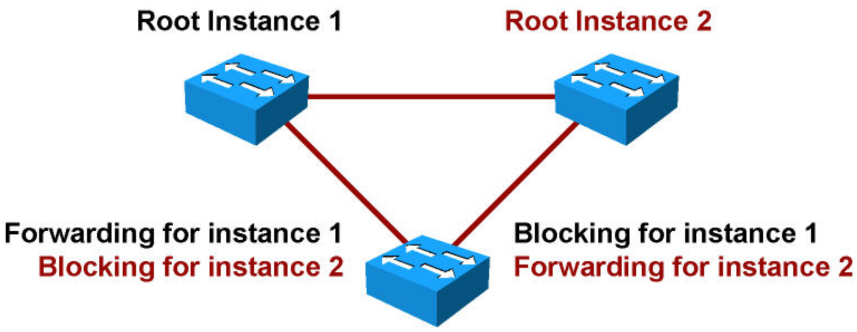


- Each switch that runs MST in the network has a single MST configuration that consists of three attributes:
  - An alphanumeric configuration name (32 bytes)
  - A configuration revision number (2 bytes)
  - A 4096-element table that associates each of the potential 4096 VLANs supported on the chassis to a given instance
- The port on B1 is at the boundary of Region A, whereas the ports on B2 and B3 are internal to Region B.

MST Bridge ID (BID)



MST Configuration Example



Instance 1 maps to VLANs 11, 21, 31
Instance 2 maps to VLANs 12, 22, 32

```
SwitchA(config)# spanning-tree mode mst
SwitchA(config)# spanning-tree mst configuration
SwitchA(config-mst)# name XYZ
SwitchA(config-mst)# revision 1
SwitchA(config-mst)# instance 1 vlan 11, 21, 31
SwitchA(config-mst)# instance 2 vlan 12, 22, 32
SwitchA(config)# spanning-tree mst 1 root primary
```

```
SwitchB(config)# spanning-tree mode mst
SwitchB(config)# spanning-tree mst configuration
SwitchB(config-mst)# name XYZ
SwitchB(config-mst)# revision 1
SwitchB(config-mst)# instance 1 vlan 11, 21, 31
SwitchB(config-mst)# instance 2 vlan 12, 22, 32
SwitchB(config)# spanning-tree mst 2 root primary
```

Verifying MST Configuration

- **`show spanning-tree mst`**
- **`show spanning-tree mst`** *`instance`*
- **`show spanning-tree mst interface`** *`interface_ID`*
- **`show spanning-tree mst`** *`instance`* **`detail`**