Chapter 2:

Implementing Spanning Tree

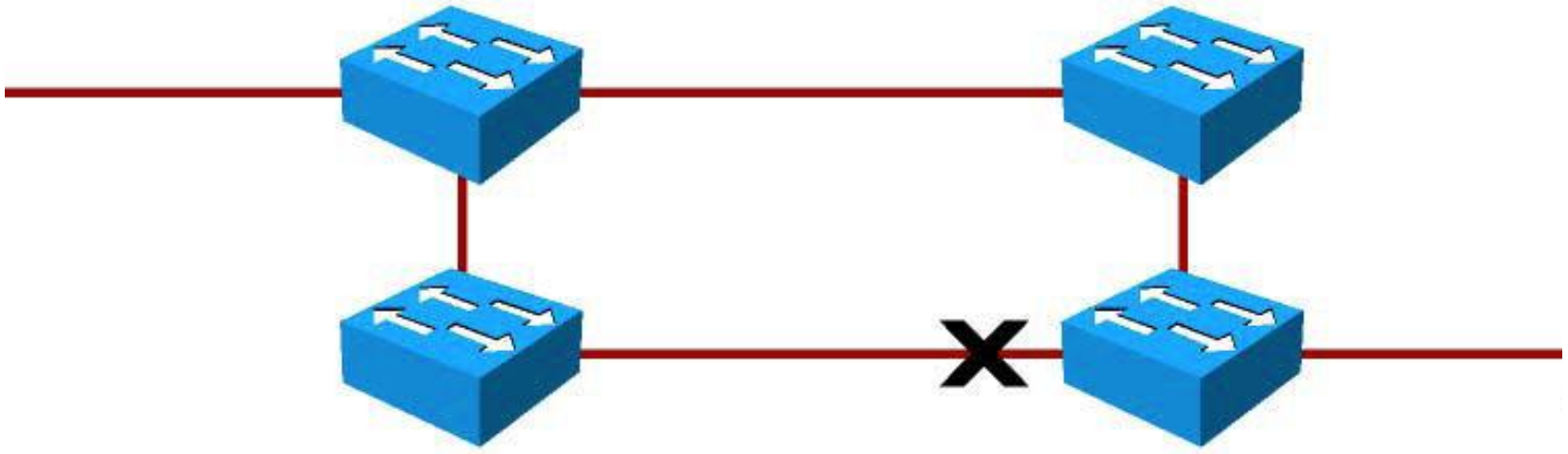
- CCNP-RS SWITCH

Chapter 2 Objectives

- Describe spanning tree protocols.
- Describe and configure RSTP.
- Describe and configure MST.
- Configure STP features to enhance resiliency and prevent forwarding loops.
- Explain recommended STP configurations and practices.
- Troubleshoot spanning tree issues.

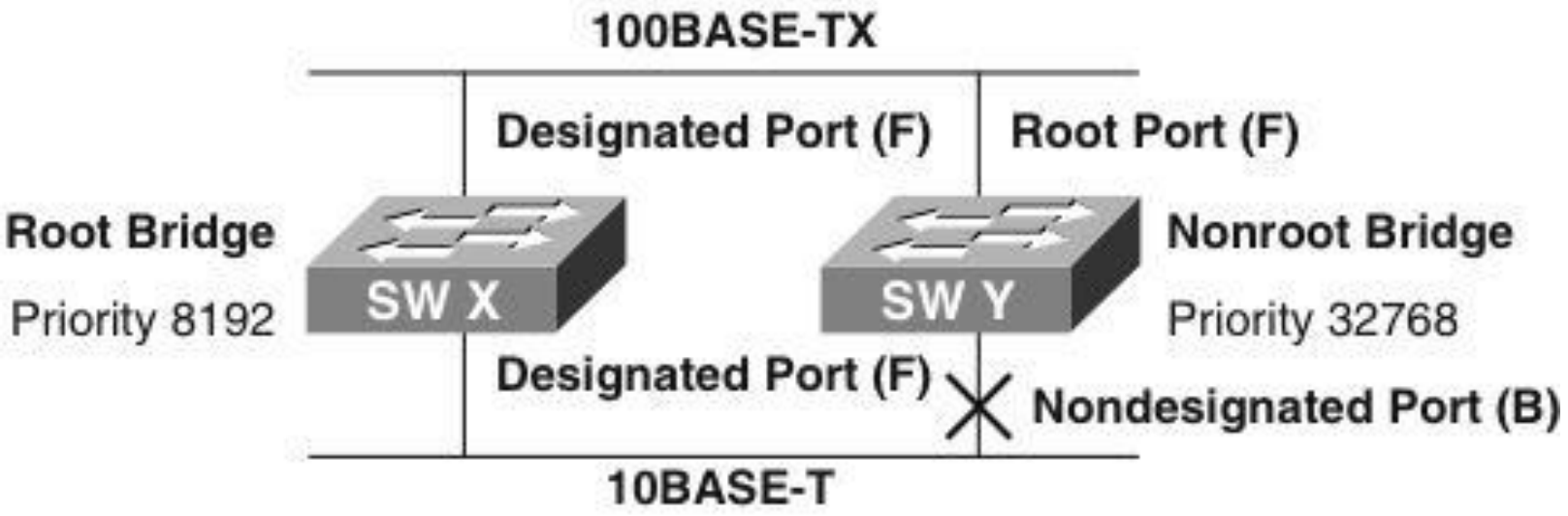
Spanning Tree Protocol Basics

Spanning Tree History

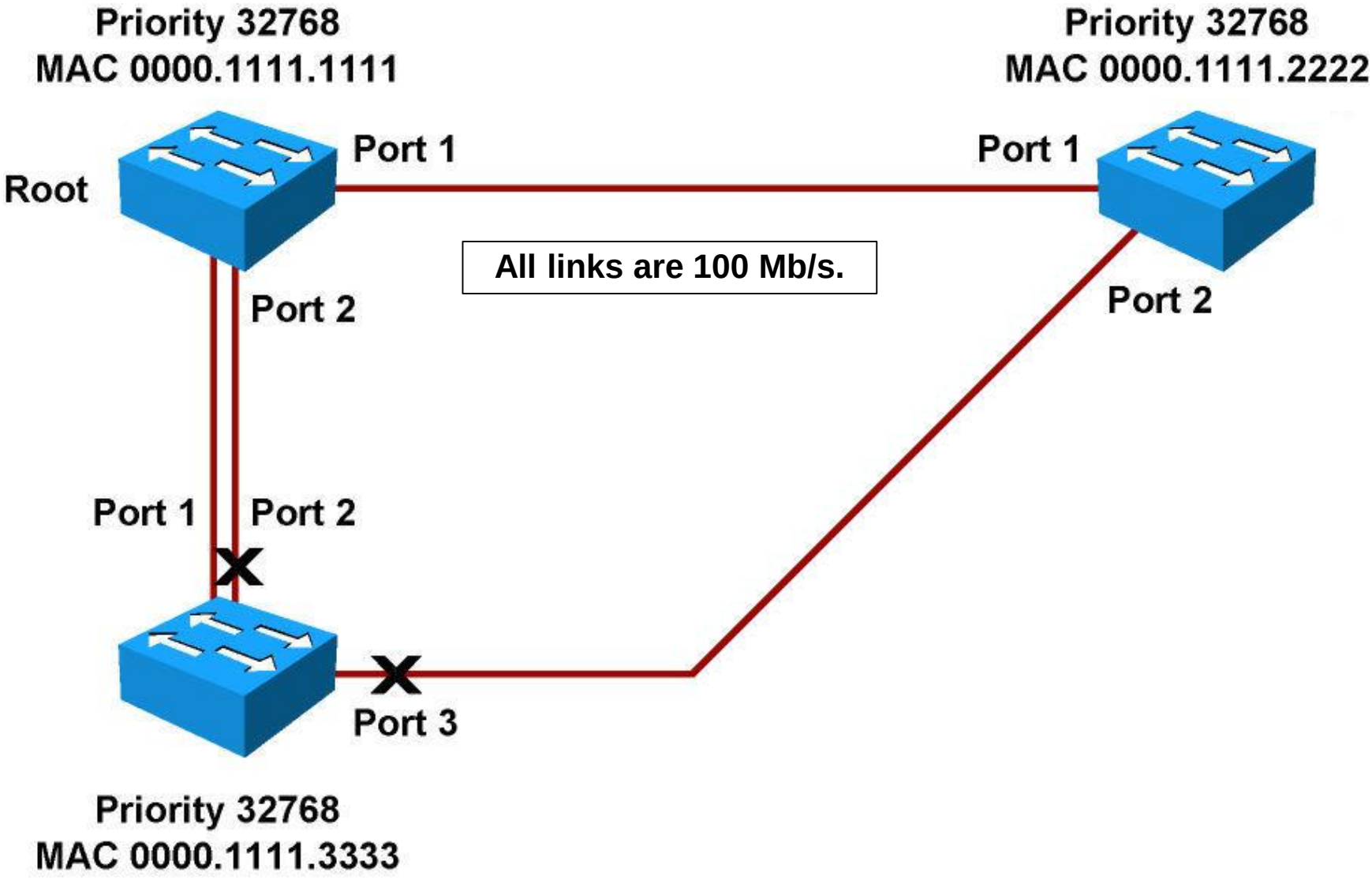


- STP was invented in 1985 by Radia Perlman at the Digital Equipment Corporation.
- In 1990, IEEE published the first standard for the protocol as 802.1D.
- Common Spanning Tree (CST) -> Cisco PVST+ -> Rapid STP (RSTP) or IEEE 802.1w -> Cisco PVRST+ -> Multiple Spanning Tree (MST) or IEEE 802.1s -> STP security enhancements

STP Operation 1 (Review from CCNA)



STP Operation 2 (Review from CCNA)



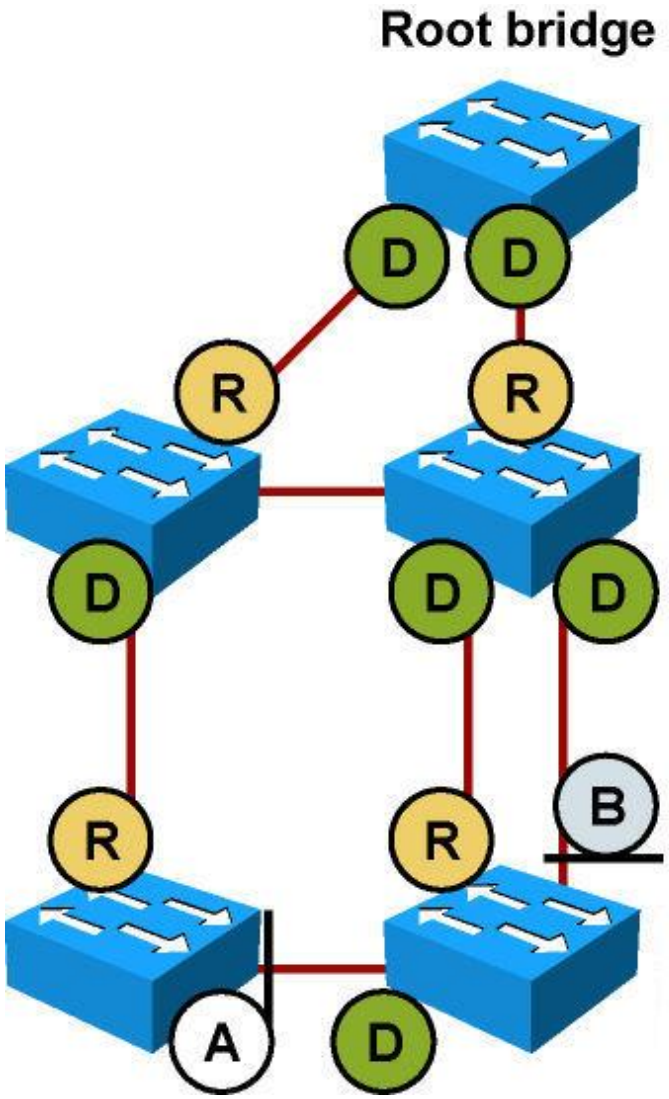
Rapid Spanning Tree Protocol

RSTP Operation – Port 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.

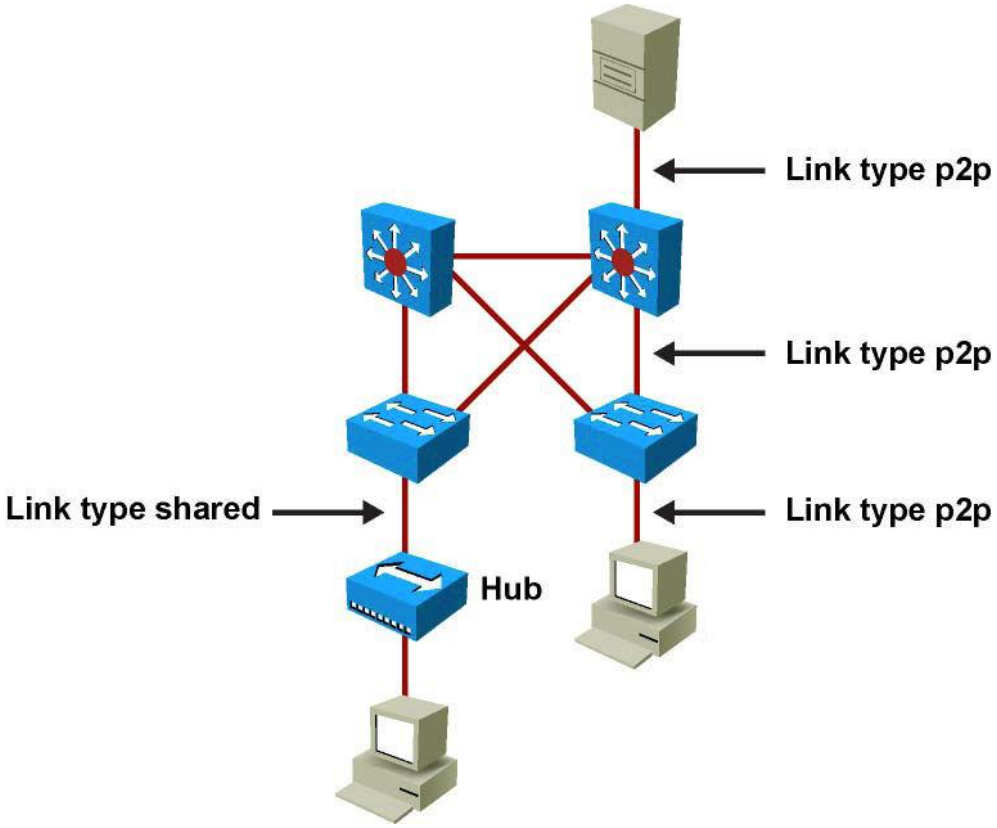
Operational Status	STP Port State	RSTP Port State	Port Included in Active Topology
Enabled	Blocking	Discarding	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

RSTP Operation – Port Roles



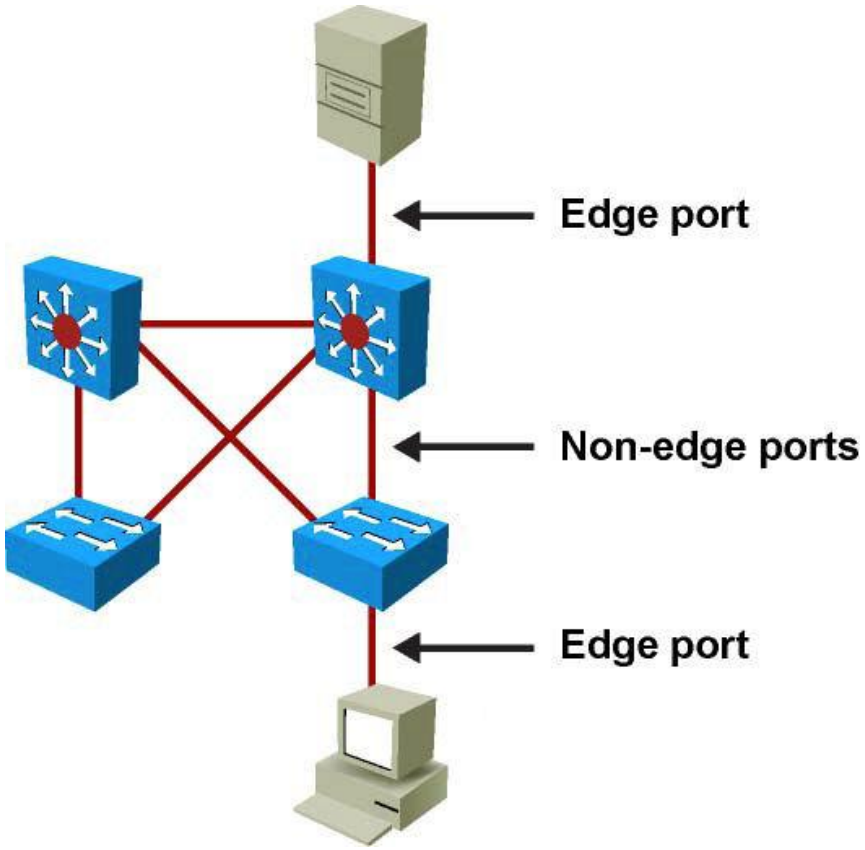
STP Port Role	RSTP Port Role	STP Port State	RSTP Port State
Root port	Root port	Forwarding	Forwarding
Designated port	Designated port	Forwarding	Forwarding
Nondesignated port	Alternate or backup port	Blocking	Discarding
Disabled	Disabled	-	Discarding
Transition	Transition	Listening Learning	Learning

RSTP Operation – Rapid Transition to Forwarding – Link Type



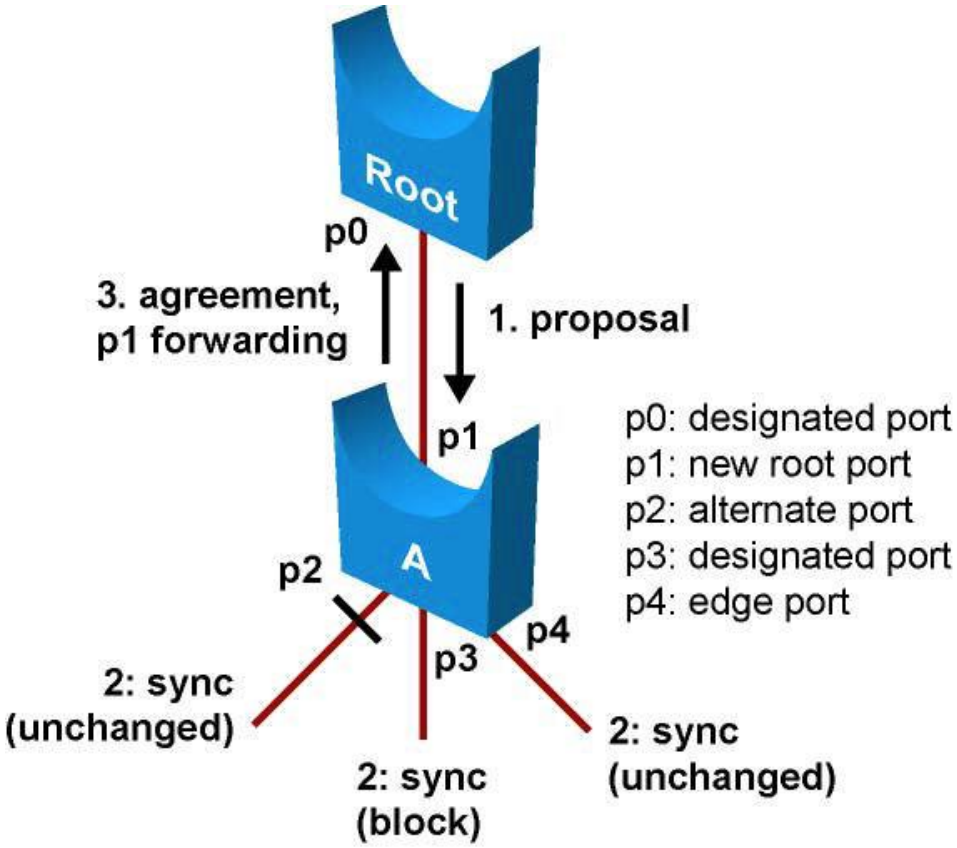
Link Type	Description
Point-to-point	Port operating in full-duplex mode. It is assumed that the port is connected to a single switch device at the other end of the link.
Shared	Port operating in half-duplex mode. It is assumed that the port is connected to shared media where multiple switches might exist.

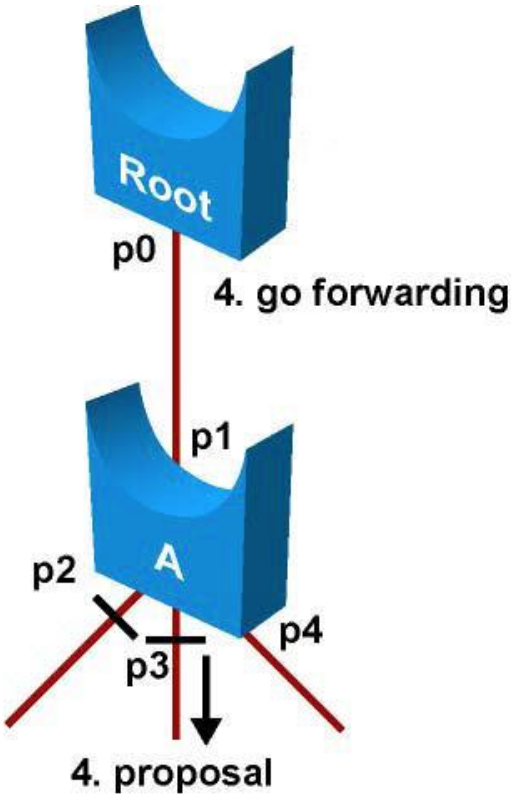
RSTP Operation – Rapid Transition to Forwarding – Edge Ports



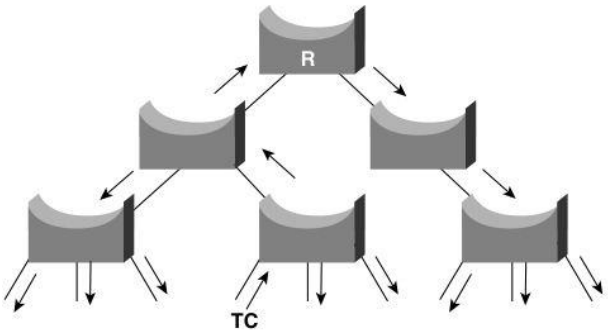
- An RSTP edge port is a switch port that is never intended to be connected to another switch device. It immediately transitions to the forwarding state when enabled.
- Neither edge ports nor PortFast-enabled ports generate topology changes when the port transitions to disabled or enabled status. Unlike PortFast, an edge port that receives a BPDU immediately loses its edge port status and becomes a normal spanning-tree port. When an edge port receives a BPDU, it generates a topology change notification (TCN).

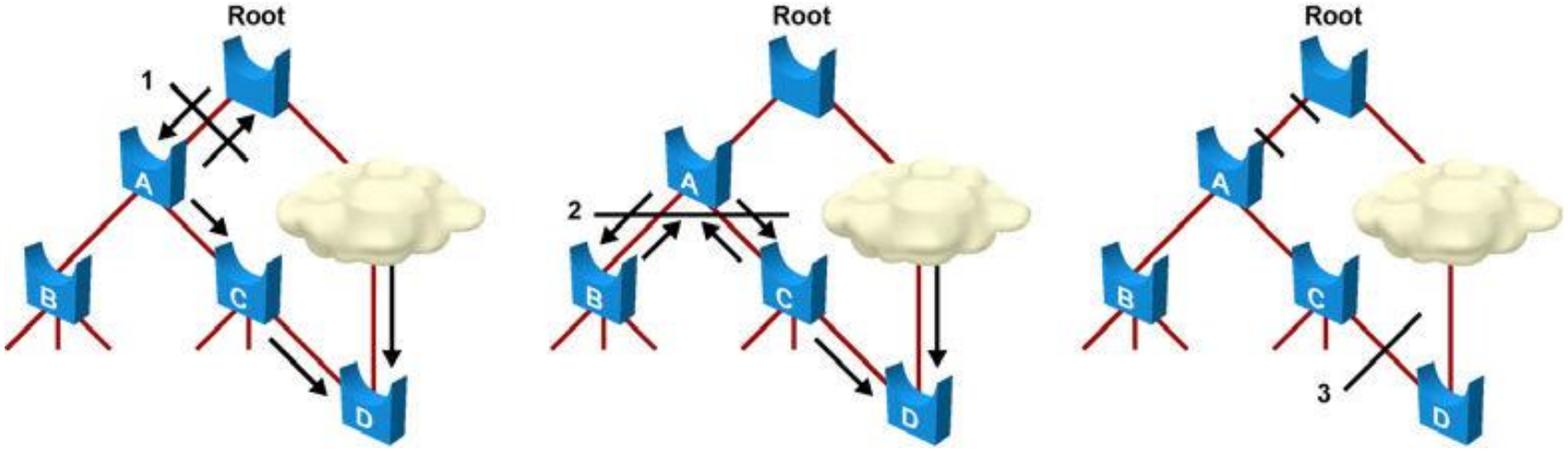
RSTP Operation – Proposal and Agreement





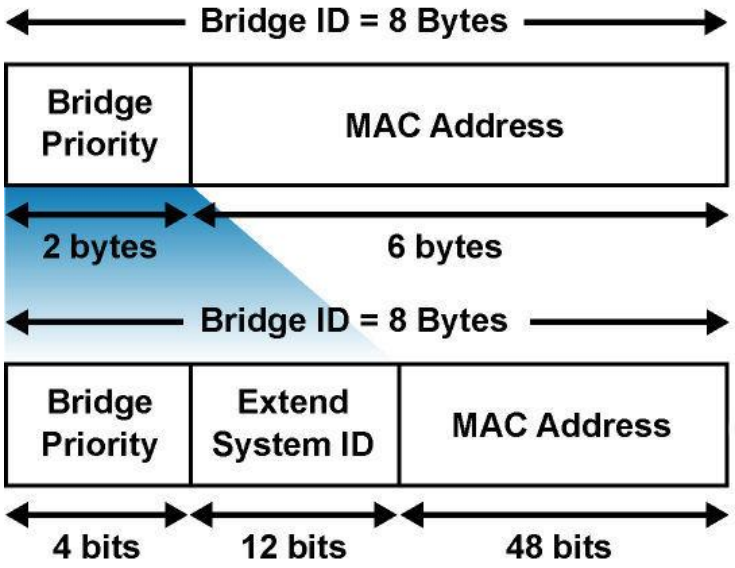
RSTP Operation – Topology Change (TC) Mechanism





- Only non-edge ports that are moving to the forwarding state cause a topology change. A port that is moving to blocking does not cause the respective bridge to generate a TC BPDU.

RSTP Operation – Bridge Identifier for PVRST+



- Only four high-order bits of the 16-bit Bridge Priority field affect the priority. Therefore, priority can be incremented only in steps of 4096, onto which are added the VLAN number. For example, for VLAN 11: If the priority is left at default, the 16-bit Priority field will hold 32768 + 11 = 32779.

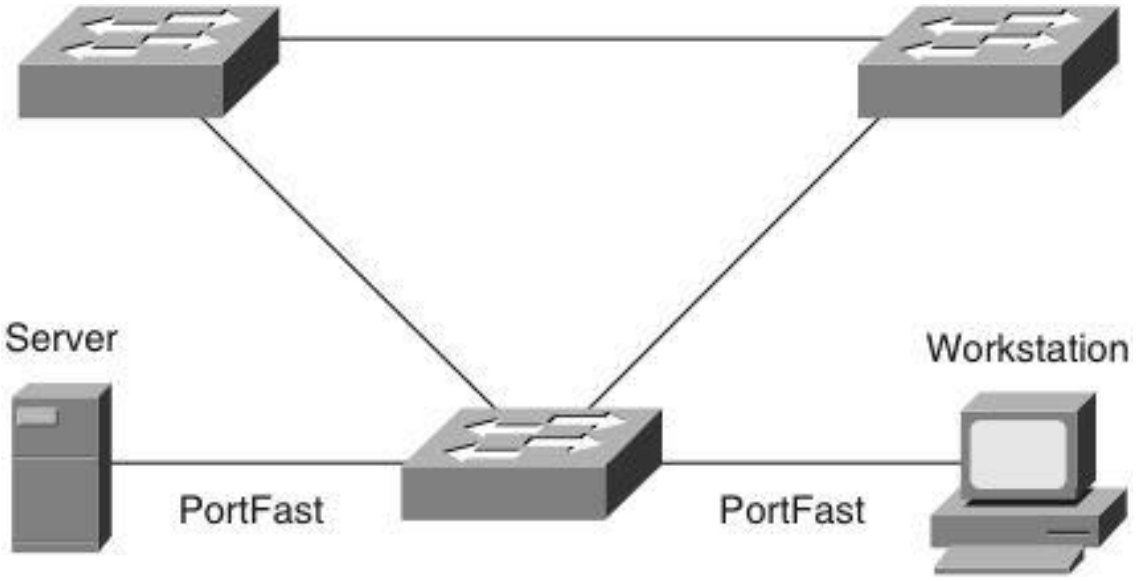
RSTP and 802.1D STP Compatibility

- RSTP can operate with 802.1D STP. However, 802.1w's fast-convergence benefits are lost when interacting with 802.1D bridges.
- Each port maintains a variable that defines the protocol to run on the corresponding segment. If the port receives BPDUs that do not correspond to its current operating mode for two times the hello time, it switches to the other STP mode.

Default STP Configuration on Cisco Switch

- PVST+
- Bridge priority 32,768 for each VLAN

Spanning Tree PortFast



- Bypass 802.1D STP listening and learning states (blocking state ⟶ forwarding state)
- Ports connected to end stations
- Prevents DHCP timeouts
- May create bridging loops if enabled on trunk port

Configuring PortFast on Access Ports

- Use the **spanning-tree portfast** interface command to enable the PortFast feature.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet 3/27
Switch(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 FastEthernet3/27 but will only
have effect when the interface is in a non-trunking mode.
Switch(config-if)# end
Switch#
Switch# show spanning-tree interface FastEthernet 3/27 portfast
VLAN0001      enabled
```

Configuring PortFast Globally

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

```
Switch(config)# spanning-tree portfast default
```

Configuring PortFast on Trunk Ports

- Use the **spanning-tree portfast trunk** interface command to enable the PortFast feature on a trunk port.

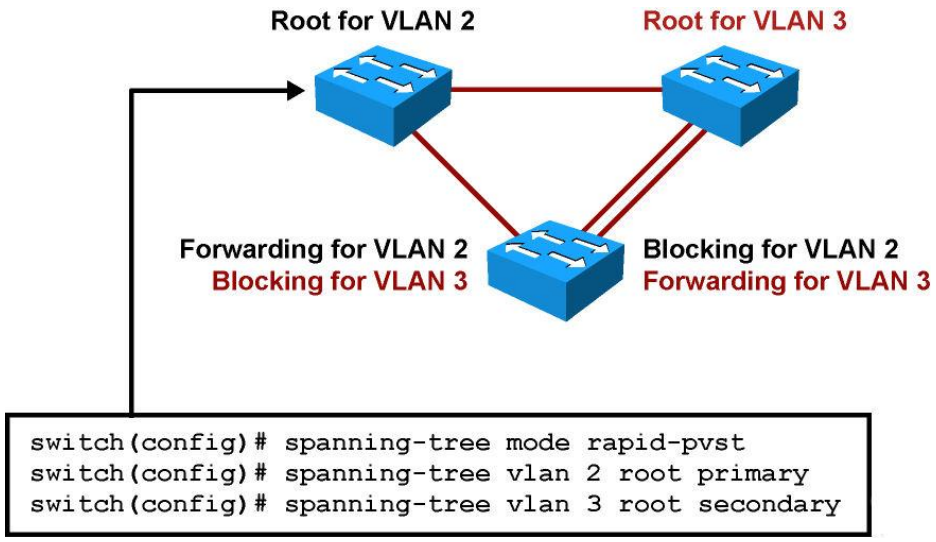
```
Switch(config)# spanning-tree portfast trunk
```

Configuring Access Port Macro

- Use the **`switchport host`** macro command on an interface connecting to an end station.

```
Switch(config-if)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled
Switch(config-if)# end
Switch#
```

Implementing PVRST+



1. Enable PVRST+ globally. PVRST+ should be configured on all switches in the broadcast domain.
2. Designate and configure a switch to be the root bridge.
3. Designate and configure a switch to be the secondary (backup) root bridge.
4. Ensure load sharing on uplinks using priority and cost parameters.
5. Verify the configuration.

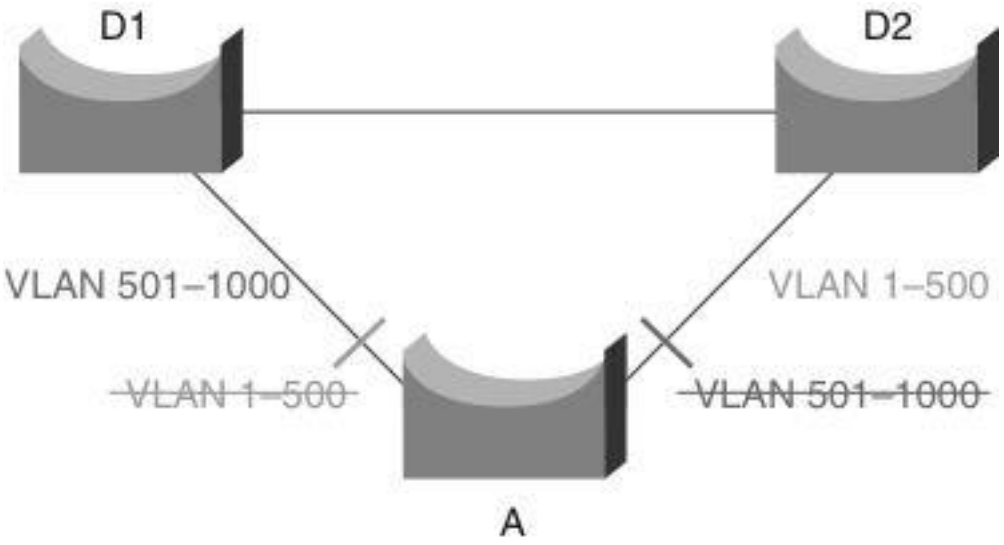
Verifying PVRST+

- The output below illustrates how to verify the RSTP configuration for VLAN2 on a nonroot switch in a topology.

```
Switch# show spanning-tree vlan 2
VLAN0002
Spanning tree enabled protocol rstp
Root ID    Priority  32768
           Address       000b.fcb5.dac0
           Cost   38
           Port   7 (FastEthernet0/7)
           Hello Time    2 sec  Max Age 20 sec       Forward Delay 15 sec
Bridge ID Priority  32770 (priority 32768 sys-id-ext 2)
           Address 0013.5f1c.e1c0
           Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
           Aging Time 300
Interface     Role   Sts    Cost   Prio.Nbr      Type
---------------    ----   ---    -------- --------   ---------------
--
Fa0/7         Root   FWD    19     128.7  P2p
Fa0/8         Root   FWD    19     128.8  P2p
```

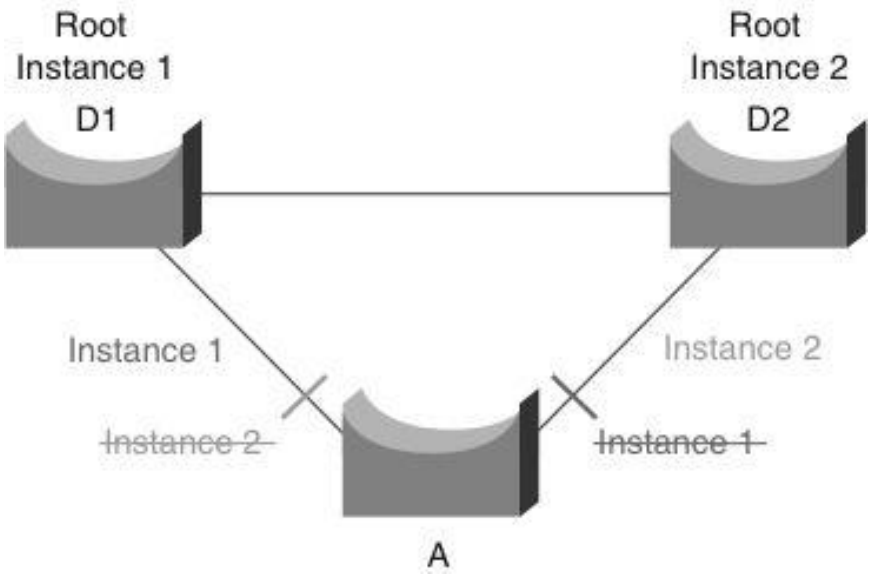
Multiple Spanning Tree

MST Motivation



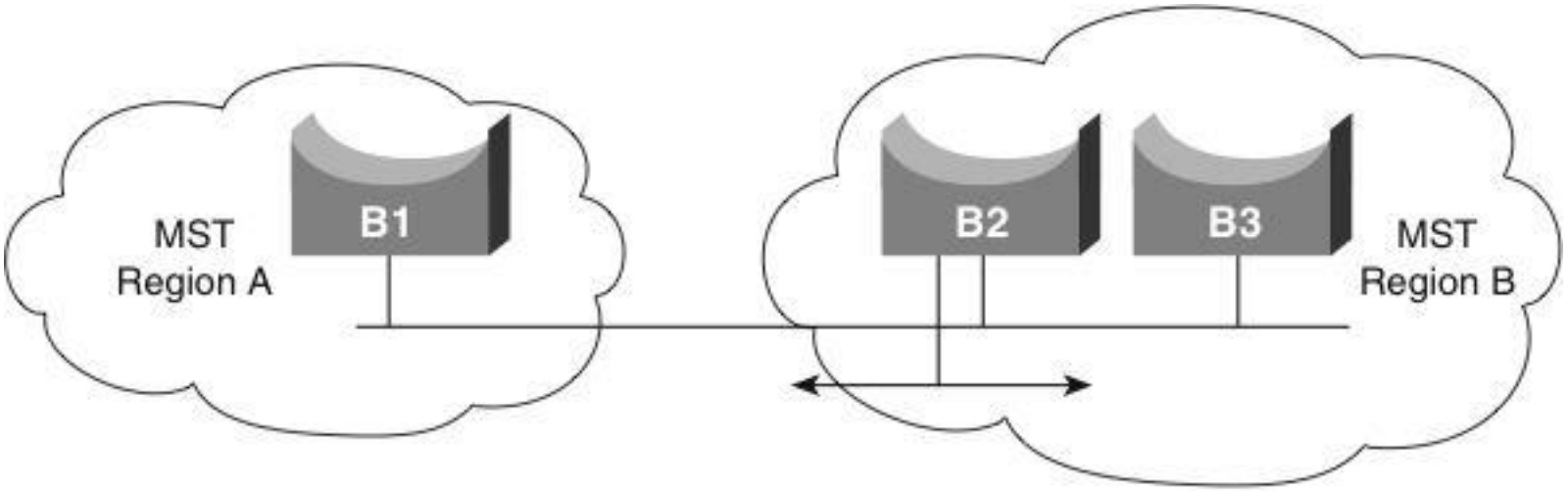
- Above: 2 links – 1000 VLANs – 2 MST instances.
- Each switch maintains only two spanning trees, reducing the need for switch resources.
- Concept extendable to 4096 VLANs: VLAN load balancing.
- MST converges faster than PVRST+ and is backward compatible with 802.1D STP and 802.1w.

MST Instances



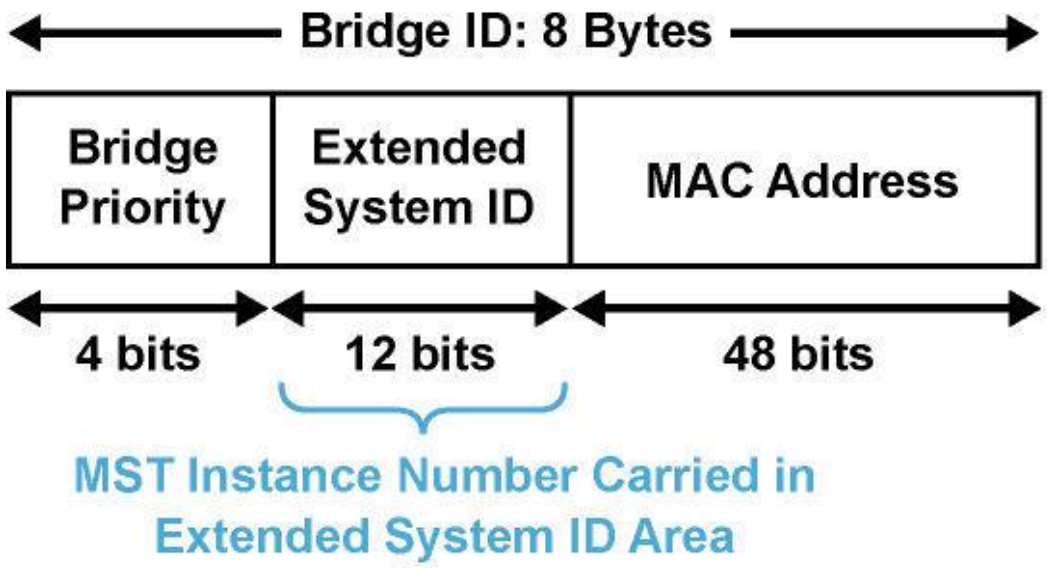
- 2 distinct STP topologies require 2 MST instances (500 per instance here).
- Load-balancing works because half of the VLANs follow each separate instance.
- Switch utilization is low because it only has to handle two instances.
- MST is the best solution for this scenario.
- Considerations: MST is 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 Use of Extended System ID



- MST carries the instance number in the 12-bit Extended System ID field of the Bridge ID.

MST Configuration

- Enable MST on switch.

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

- Enter MST configuration submode.

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

- Display current MST configuration.

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

- Name MST instance.

```
Switch(config-mst)# name name
```

- Set the 16-bit MST revision number. It is not incremented automatically when you commit a new MST configuration.

```
Switch(config-mst)# revision revision_number
```

MST Configuration (cont)

- Map VLANs to MST instance.

  Switch(config-mst)# **instance** *instance_number* **vlan** *vlan_range*

- Display new MST configuration to be applied.

  Switch(config-mst)# **show pending**

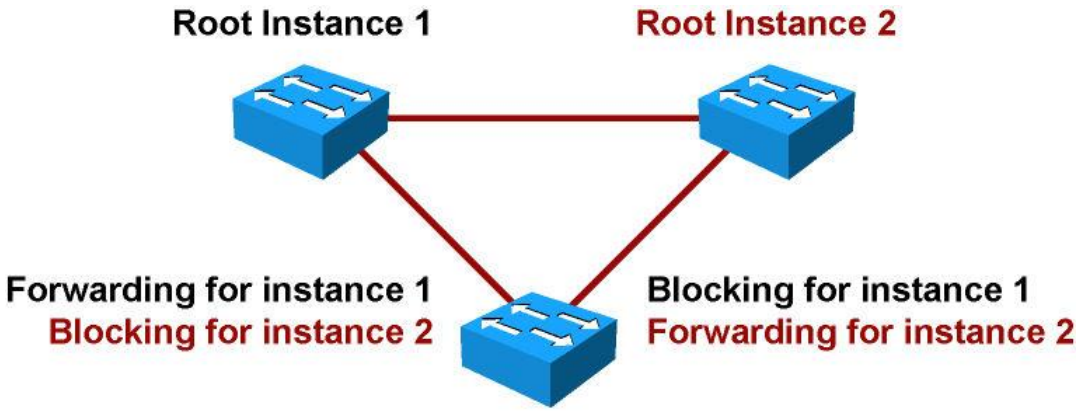
- Apply configuration and exit MST configuration submode.

  Switch(config-mst)# **exit**

- Assign root bridge for MST instance. This syntax makes the switch root primary or secondary (only active if primary fails). It sets primary priority to 24576 and secondary to 28672.

  Switch(config)# **spanning-tree mst** *instance_number* **root** *primary* | *secondary*

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 Example (1)

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# spanning-tree mode mst
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# show current
Current MST configuration
Name []
Revision 0
Instance Vlans mapped
-------- ---------------------------------------------------------------
0 1-4094
-------------------------------------------------------------------------
Switch(config-mst)# name cisco
Switch(config-mst)# revision 1
Switch(config-mst)# instance 1 vlan 1-10
Switch(config-mst)# show pending
Pending MST configuration
Name [cisco]
Revision 1
Instance Vlans mapped
-------- ---------------------------------------------------------------
0 11-4094
1 1-10
Switch(config-mst)# end
```

Verifying MST Configuration Example (2)

```
Switch# show spanning-tree mst
###### MST00 vlans mapped: 5-4094
Bridge            address 0009.e845.6480     priority 32768 (32768 sysid 0)
Root              this switch for CST and IST
Configured        hello time 2, forward delay 15, max age 20, max hops 20

Interface         Role     Sts      Cost       Prio.Nbr   Type
----------------  ----     ---      ---------  --------   -------
Fa3/24            Desg     FWD      2000000    128.152    Shr
Fa3/32            Desg     FWD      200000     128.160    P2p
Fa3/42            Back     BLK      200000     128.170    P2p
###### MST01      vlans mapped: 1-2
Bridge            address 0009.e845.6480     priority 32769 (32768 sysid 1)
Root              this switch for MST01
Interface         Role     Sts      Cost       Prio.Nbr   Type
----------------  ----     ---      ---------  --------   -------
Fa3/24            Desg     FWD      2000000    128.152    Shr
Fa3/32            Desg     FWD      200000     128.160    P2p
Fa3/42            Back     BLK      200000     128.170    P2p
###### MST02 vlans mapped: 3-4
Bridge            address 0009.e845.6480     priority 32770 (32768 sysid 2)
Root              this switch for MST02
Interface         Role     Sts      Cost       Prio.Nbr   Type
----------------  ----     ---      ---------  --------   -------
Fa3/24            Desg     FWD      2000000    128.152    Shr
```

Verifying MST Configuration Example (3)

```
Switch# show spanning-tree mst 1
###### MST01    vlans mapped: 1-2
Bridge          address 0009.e845.6480 priority 32769 (32768 sysid 1)
Root            this switch for MST01
Interface               Role    Sts     Cost    Prio.Nbr        Type
---------------         ----    ---     ------  --------        -----------------
Fa3/24                  Desg    FWD     2000000 128.152         Shr
Fa3/32                  Desg    FWD     200000  128.160         P2p
Fa3/42                  Back    BLK     200000  128.170         P2p
```

Verifying MST Configuration Example (4)

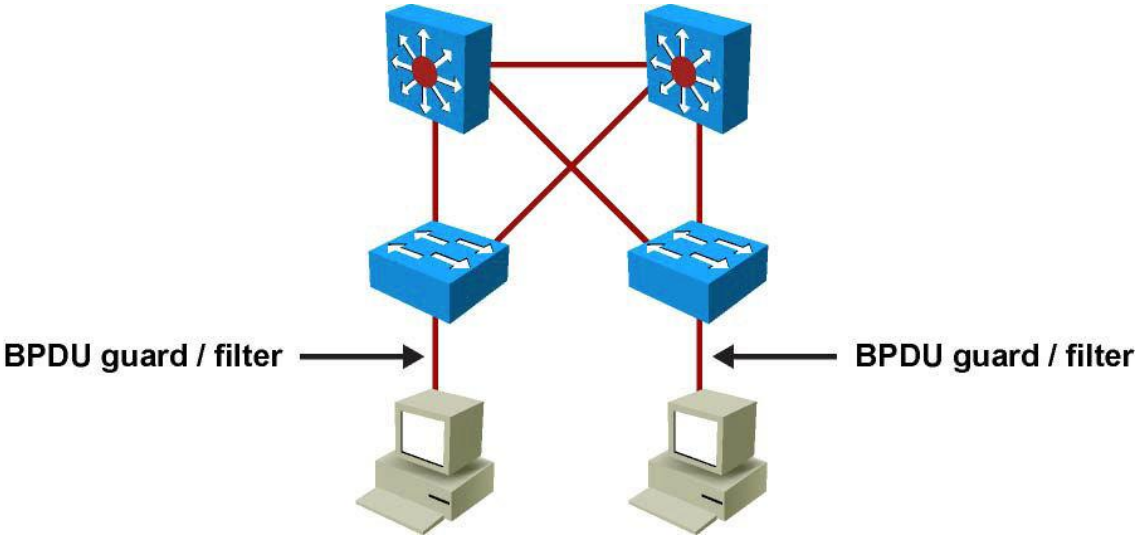
```
Switch# show spanning-tree mst interface FastEthernet 3/24

FastEthernet3/24 of MST00 is designated forwarding
Edge port: no           (default)       port guard : none       (default)
Link type: shared       (auto)          bpdu filter: disable    (default)
Boundary : internal                     bpdu guard : disable    (default)
Bpdus sent 81, received 81

Instance        Role    Sts     Cost    Prio.Nbr        Vlans mapped
--------        ----    ---     ------- --------        ------------------------
0                       Desg    FWD     2000000 128.152         5-4094
1                       Desg    FWD     2000000 128.152         1-2
2                       Desg    FWD     2000000 128.152         3-4
```

Verifying MST Configuration Example (5)

```
Switch# show spanning-tree mst 1 detail
###### MST01              vlans mapped: 1-2
Bridge                   address 0009.e845.6480 priority 32769 (32768 sysid 1)
Root                     this switch for MST01
FastEthernet3/24 of MST01 is designated forwarding
Port info                port id 128.152 priority 128 cost 2000000
Designated root                  address 0009.e845.6480 priority 32769 cost 0
Designated bridge        address 0009.e845.6480 priority 32769 port id 128.152
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent755, received 0
FastEthernet3/32 of MST01 is designated forwarding
Port info                port id 128.160 priority 128 cost 200000
Designated root                  address 0009.e845.6480 priority 32769 cost 0
Designated bridge        address 0009.e845.6480 priority 32769 port id 128.160
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 769, received 1
FastEthernet3/42 of MST01 is backup blocking
Port info                port id 128.170 priority 128 cost 200000
Designated root                  address 0009.e845.6480 priority 32769 cost 0
Designated bridge        address 0009.e845.6480 priority 32769 port id 128.160
Timers: message expires in 5 sec, forward delay 0, forward transitions 0
Bpdus (MRecords) sent 1, received 769
```

Understanding Spanning Tree Enhancements

Spanning Tree Enhancements



- **BPDU guard**: Prevents accidental connection of switching devices to PortFast-enabled ports. Connecting switches to PortFast-enabled ports can cause Layer 2 loops or topology changes.
- **BPDU filtering:** Restricts the switch from sending unnecessary BPDUs out access ports.
- **Root guard**: Prevents switches connected on ports configured as access ports from becoming the root switch.
- **Loop guard**: Prevents root ports and alternate ports from moving to forwarding state when they stop receiving BPDUs.

BPDU Guard

- BPDU Guard puts an interface configured for STP PortFast in the err-disable state upon receipt of a BPDU. BPDU guard disables interfaces as a preventive step to avoid potential bridging loops.
- BPDU guard shuts down PortFast-configured interfaces that receive BPDUs, rather than putting them into the STP blocking state (the default behavior). In a valid configuration, PortFast-configured interfaces should not receive BPDUs. Reception of a BPDU by a PortFast-configured interface signals an invalid configuration, such as connection of an unauthorized device.
- BPDU guard provides a secure response to invalid configurations, because the administrator must manually re-enable the err-disabled interface after fixing the invalid configuration. It is also possible to set up a time-out interval after which the switch automatically tries to re-enable the interface. However, if the invalid configuration still exists, the switch err-disables the interface again.

BPDU Guard Configuration

- To enable BPDU guard globally, use the command:

  spanning-tree portfast bpduguard default

- To enable BPDU guard on a port, use the command:

  spanning-tree bpduguard enable

- BPDU guard logs messages to the console:

```
2009 May 12 15:13:32 %SPANTREE-2-
RX_PORTFAST:Received BPDU on PortFast enable port.
Disabling 2/1
2009 May 12 15:13:32 %PAGP-5-PORTFROMSTP:Port 2/1
left bridge port 2/1
```

BPDU Guard Configuration Example

```
Switch(config)# spanning-tree portfast edge bpduguard default
Switch(config)# end
Switch# show spanning-tree summary totals
Root bridge for: none.
PortFast BPDU Guard is enabled
Etherchannel misconfiguration guard is enabled
UplinkFast is disabled
BackboneFast is disabled
Default pathcost method used is short
Name          Blocking Listening Learning Forwarding STP Active
------------ -------- --------- -------- ---------- ---------
34 VLANs      0        0           0       36          36
```

BPDU Filtering

- BPDU filtering prevents a Cisco switch from sending BPDUs on PortFast-enabled interfaces, preventing unnecessary BPDUs from being transmitted to host devices.
- BPDU guard has no effect on an interface if BPDU filtering is enabled.
- When enabled globally, BPDU filtering has these attributes:
  - It affects all operational PortFast ports on switches that do not have BPDU filtering configured on the individual ports.
  - If BPDUs are seen, the port loses its PortFast status, BPDU filtering is disabled, and 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 PortFast BPDU filtering are disabled.
- When enabled on an interface, BPDU filtering has these attributes:
  - It ignores all BPDUs received.
  - It sends no BPDUs.

BPDU Filtering Configuration

- To enable BPDU filtering globally, use the command:
  `spanning-tree portfast bpdufilter default`
- To enable BPDU guard on a port, use the command:
  `spanning-tree bpdufilter enable`

Verifying BPDU Filtering Configuration (1)

- PortFast BPDU filtering status:

```
Switch# show spanning-tree summary
Switch is in pvst mode
Root bridge for: none
Extended system ID                  is enabled
Portfast Default                    is disabled
PortFast BPDU Guard Default         is disabled
Portfast BPDU Filter Default        is disabled
Loopguard Default                   is disabled
EtherChannel misconfig guard        is enabled
UplinkFast                          is disabled
BackboneFast                        is disabled
Configured Pathcost method used is short
Name    Blocking        Listening       Learning        Forwarding      STP
Active
------- ----            --------        -------         --------        ---------
VLAN0001  2             0               0               6                8
------- ----            --------        -------         --------        ---------
1 vlan    2             0               0               6                8
```

Verifying BPDU Filtering Configuration (2)

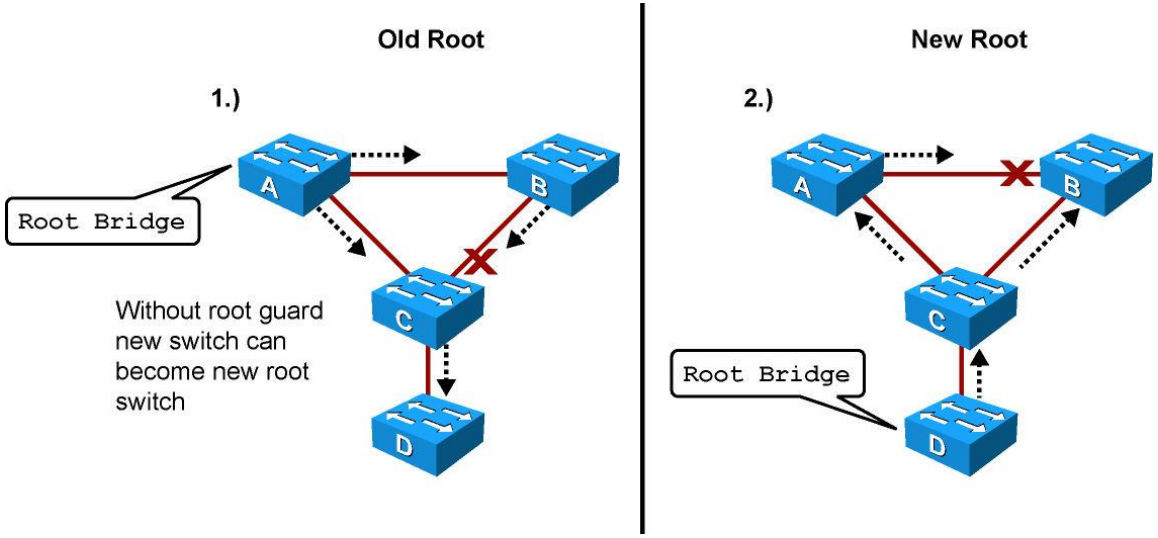
- Verifying PortFast BPDU filtering on a specific port:

```
Switch# show spanning-tree interface fastEthernet 4/4 detail

Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   BPDU:sent 0, received 0
```

Root Guard

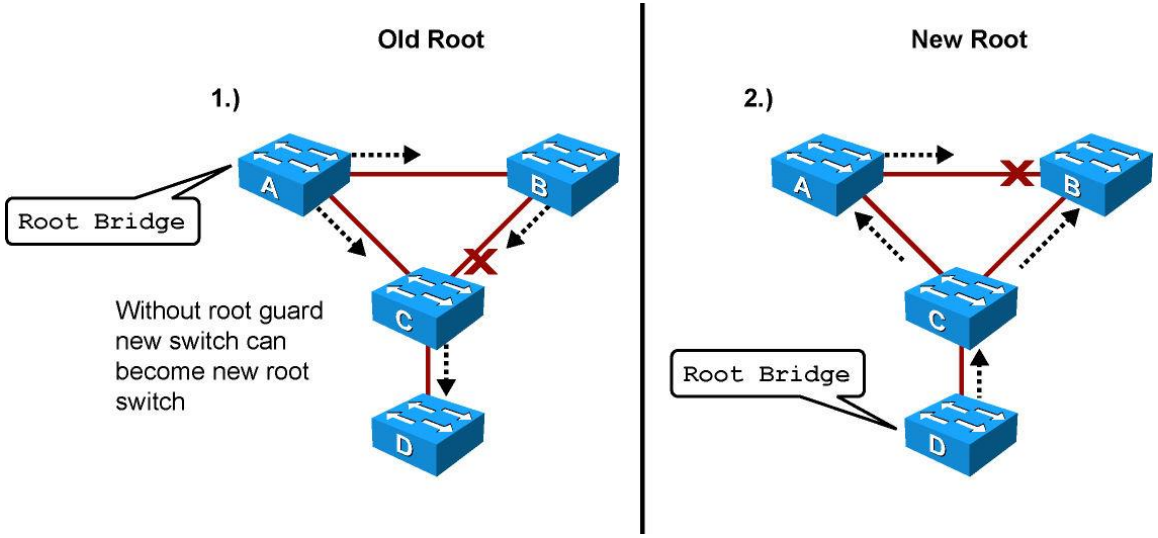
- Root guard is useful in avoiding Layer 2 loops during network anomalies. The Root guard feature forces an interface to become a designated port to prevent surrounding switches from becoming root bridges.
- Root guard-enabled ports are forced to be designated ports. If the bridge receives superior STP BPDUs on a Root guard-enabled port, the port moves to a root-inconsistent STP state, which is effectively equivalent to the STP listening state, and the switch does not forward traffic out of that port. As a result, this feature enforces the position of the root bridge.

Root Guard Motivation



- Switches A and B comprise the core of the network. Switch A is the root bridge.
- Switch C is an access layer switch. When Switch D is connected to Switch C, it begins to participate in STP. If the priority of Switch D is 0 or any value lower than that of the current root bridge, Switch D becomes the root bridge.
- Having Switch D as the root causes the Gigabit Ethernet link connecting the two core switches to block, thus causing all the data to flow via a 100-Mbps link across the access layer. This is obviously a terrible outcome.

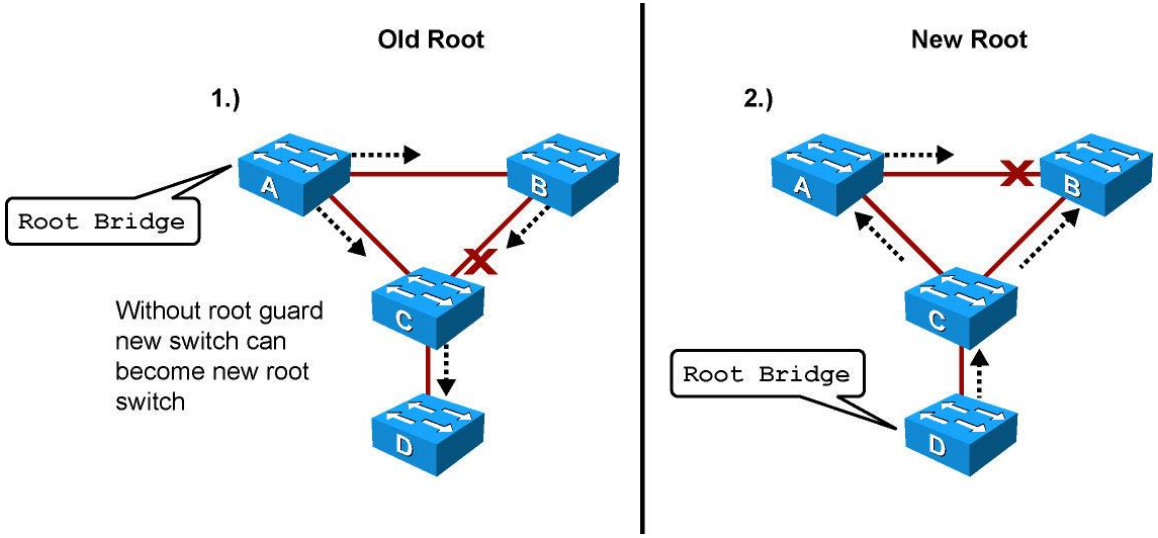
Root Guard Operation



- After the root guard feature is enabled on a port, the switch does not enable that port to become an STP root port.
- Cisco switches log the following message when a root guard–enabled port receives a superior BPDU:

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

Root Guard Operation



- The current design recommendation is to enable root guard on all access ports so that a root bridge is not established through these ports.
- In this configuration, Switch C blocks the port connecting to Switch D when it receives a superior BPDU. The port transitions to the root-inconsistent STP state. No traffic passes through the port while it is in root-inconsistent state.
- When Switch D stops sending superior BPDUs, the port unblocks again and goes through regular STP transition of listening and learning, and eventually to the forwarding state. Recovery is automatic; no intervention is required.

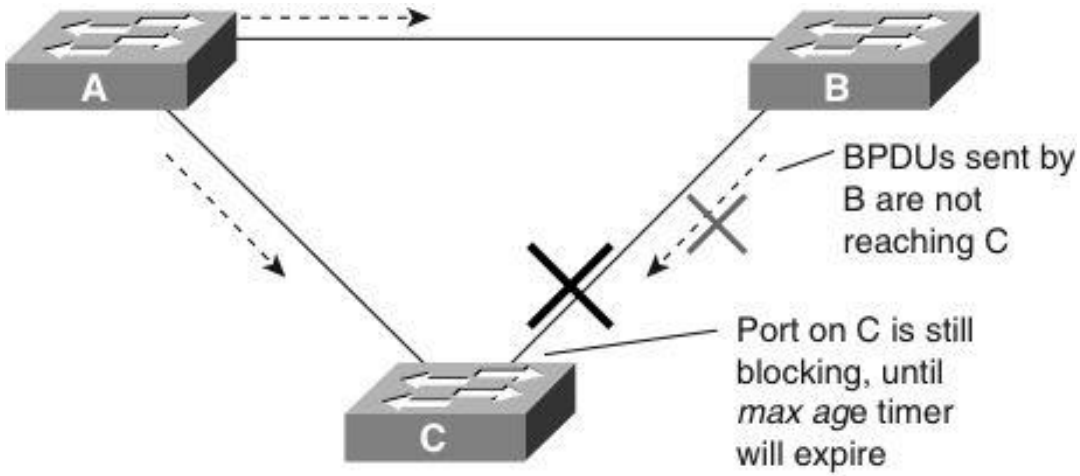
Root Guard Configuration

```
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
```

Verifying Root Guard Configuration

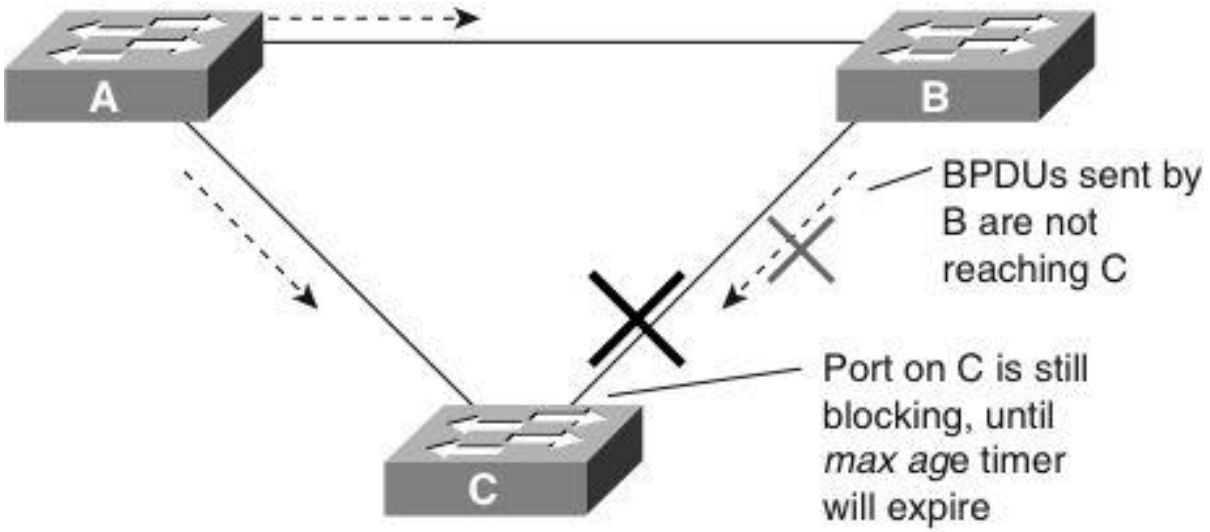
```
Switch# show spanning-tree inconsistentports
Name                 Interface              Inconsistency
-------------------  ---------------------- ------------------
VLAN0001             FastEthernet3/1        Port Type Inconsistent
VLAN0001             FastEthernet3/2        Port Type Inconsistent
VLAN1002             FastEthernet3/1        Port Type Inconsistent
VLAN1002             FastEthernet3/2        Port Type Inconsistent
Number of inconsistent ports (segments) in the system :4
```

Loop Guard



- The Loop Guard STP feature improves the stability of Layer 2 networks by preventing bridging loops.
- In STP, switches rely on continuous reception or transmission of BPDUs, depending on the port role. A designated port transmits BPDUs whereas a nondesignated port receives BPDUs.
- Bridging loops occur when a port erroneously transitions to forwarding state because it has stopped receiving BPDUs.
- Ports with loop guard enabled do an additional check before transitioning to forwarding state. If a nondesignated port stops receiving BPDUs, the switch places the port into the STP *loop-inconsistent* blocking state.
- If a switch receives a BPDU on a port in the loop-inconsistent STP state, the port transitions through STP states according to the received BPDU. As a result, recovery is automatic, and no manual intervention is necessary.

Loop Guard Messages



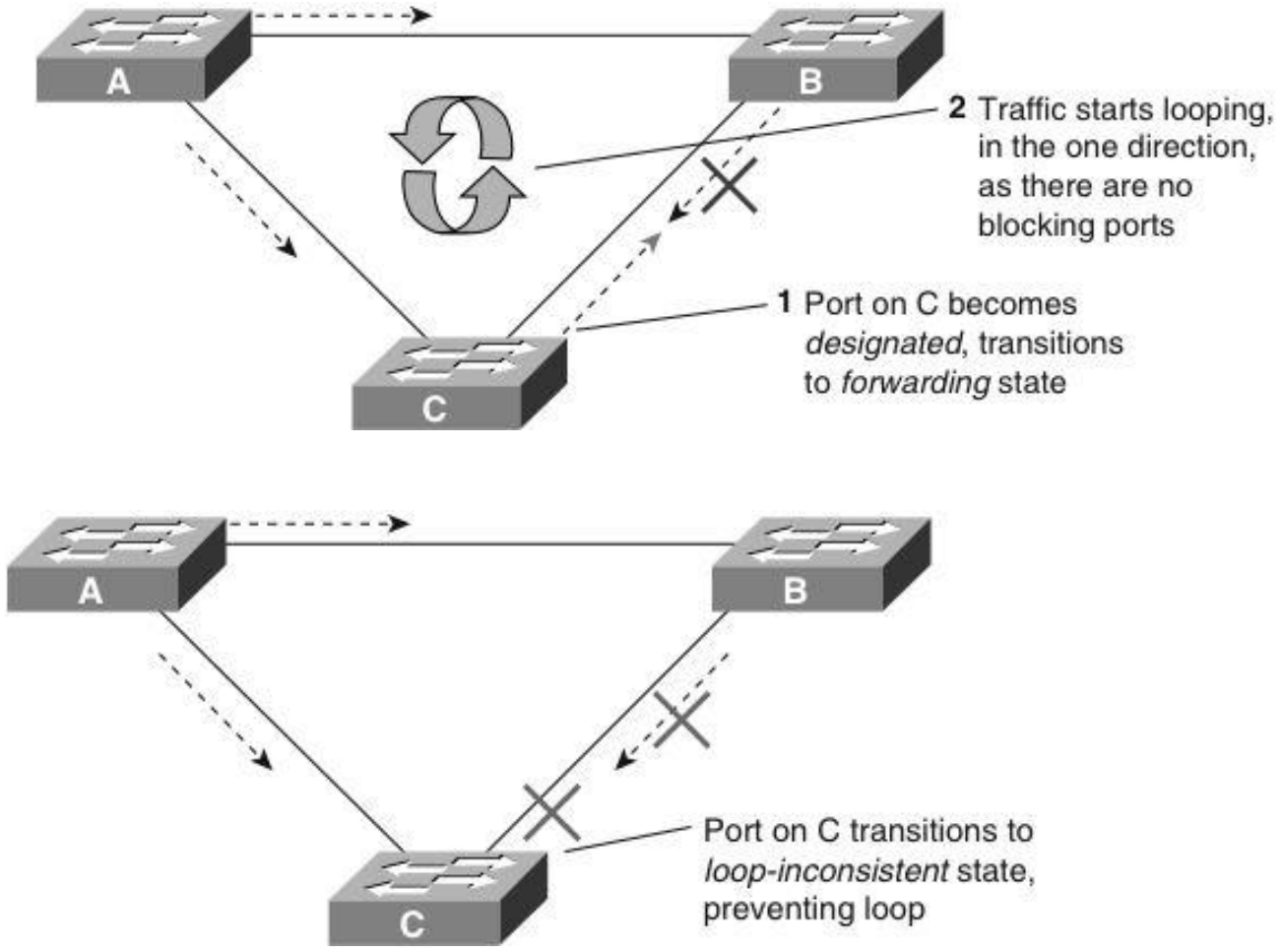
- When the Loop Guard feature places a port into the loop-inconsistent blocking state, the switch logs the following message:

```
SPANTREE-2-LOOPGUARDBLOCK: No BPDUs were received on port 3/2
in vlan 3.

Moved to loop-inconsistent state.
```

- After recovery, the switch logs the following message:

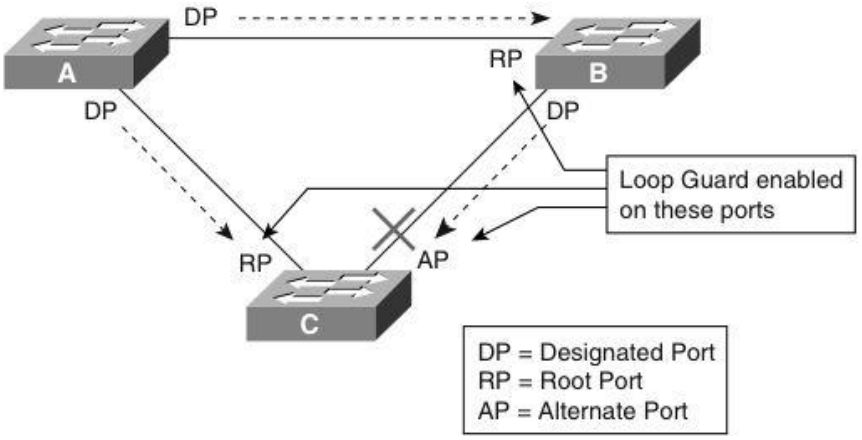
```
SPANTREE-2-LOOPGUARDUNBLOCK: port 3/2 restored in vlan 3.
```

Loop Guard Operation



Loop Guard Configuration Considerations

- Configure Loop Guard on a per-port basis, although the feature blocks inconsistent ports on a per-VLAN basis; for example, on a trunk port, if BPDUs are not received for only one particular VLAN, the switch blocks only that VLAN (that is, moves the port for that VLAN to the loop-inconsistent STP state). In the case of an EtherChannel interface, the channel status goes into the inconsistent state for all the ports belonging to the channel group for the particular VLAN not receiving BPDUs.
- Enable Loop Guard on all nondesignated ports. Loop guard should be enabled on root and alternate ports for all possible combinations of active topologies.
- Loop Guard is disabled by default on Cisco switches.



Loop Guard Configuration

- Use the following interface-level configuration command to enable Loop Guard:

```
Switch(config-if)# spanning-tree guard loop
```

- If Loop Guard is enabled globally, the switch enables Loop Guard only on ports considered to be point-to-point links (full-duplex links).
- The global configuration can be overridden on a per-port basis. To enable Loop Guard globally, use the following global configuration command:

```
Switch(config)# spanning-tree loopguard default
```

Verifying Loop Guard Configuration

- To verify Loop Guard status on an interface, issue the following :

```
Switch(config-if)# spanning-tree guard loop
```

- If Loop Guard is enabled globally, the switch enables Loop Guard only on ports considered to be point-to-point links (full-duplex links). The global configuration can be overridden on a per-port basis. To enable Loop Guard globally, use the following global configuration command:

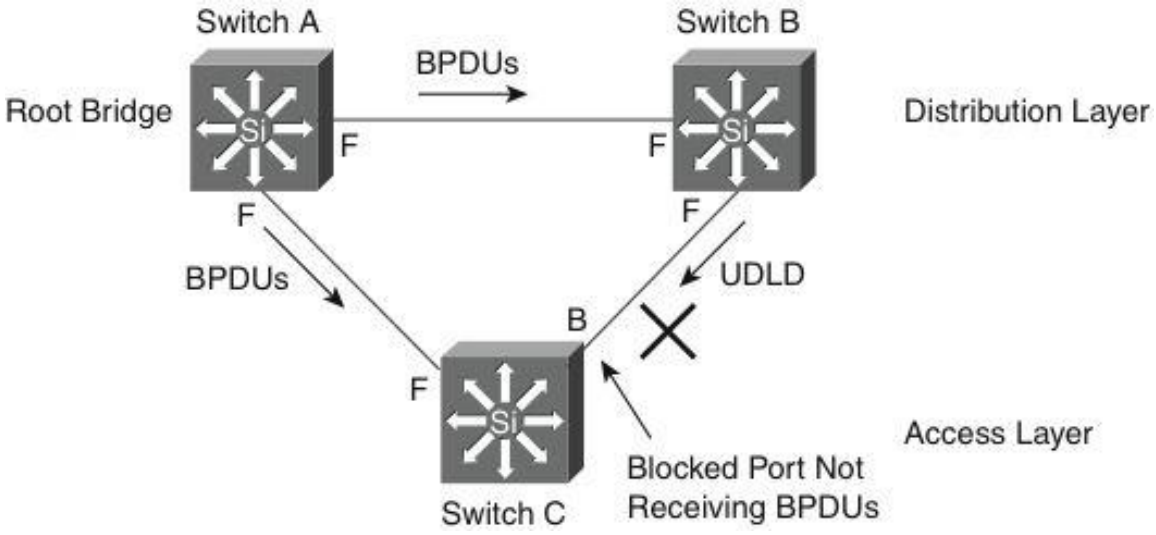
```
Switch(config)# spanning-tree loopguard default
```

Verifying Loop Guard Configuration

- To verify Loop Guard status on an interface, issue the command **`show spanning-tree interface`** *`interface-id`* **`detail`**.

```
Switch# show spanning-tree interface FastEthernet 3/42 detail
  Port 170 (FastEthernet3/42) of VLAN0001 is blocking
  Port path cost 19, Port priority 128, Port Identifier 128.170.
  Designated root has priority 8193, address 0009.e845.6480
  Designated bridge has priority 8193, address 0009.e845.6480
  Designated port id is 128.160, designated path cost 0
  Timers: message age 1, forward delay 0, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default
  Loop guard is enabled on the port
  BPDU: sent 1, received 4501
```

Unidirectional Link Detection (UDLD)



- The link between Switches B and C becomes unidirectional. Switch B can receive traffic from Switch C, but Switch C cannot receive traffic from Switch B.
- On the segment between Switches B and C, Switch B is the designated bridge sending the root BPDUs and Switch C expects to receive the BPDUs.
- Switch C waits until the max-age timer (20 seconds) expires before it takes action. When this timer expires, Switch C moves through the listening and learning states and then to the forwarding state. At this moment, both Switch B and Switch C are forwarding to each other and there is no blocking port in the network.

UDLD Modes

- **Normal Mode** –UDLD detects unidirectional links due to misconnected interfaces on fiber-optic connections. UDLD changes the UDLD-enabled port to an undetermined state if it stops receiving UDLD messages from its directly connected neighbor.
- **Aggressive Mode** – (Preferred) When a port stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port state changes to the err-disable state. Aggressive mode UDLD detects unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and due to misconnected interfaces on fiber-optic links.

UDLD Configuration

- UDLD is disabled on all interfaces by default.
- The **udld** global configuration command affects fiber-optic interfaces only.
  - **udld enable** enables UDLD normal mode on all fiber interfaces.
  - **udld aggressive** enables UDLD aggressive mode on all fiber interfaces.
- The **udld port** interface configuration command can be used for twisted-pair and fiber interfaces.
  - To enable UDLD in normal mode, use the **udld port** command. To enable UDLD in aggressive mode, use the **udld port aggressive**.
  - Use the **no udld port** command on fiber-optic ports to return "control" of UDLD to the **udld enable** global configuration command or to disable UDLD on nonfiber-optic ports.
  - Use the **udld port aggressive** command on fiber-optic ports to override the setting of the **udld enable** or **udld aggressive** global configuration command. Use the **no** form on fiber-optic ports to remove this setting and to return control of UDLD enabling to the **udld** global configuration command or to disable UDLD on nonfiber-optic ports.

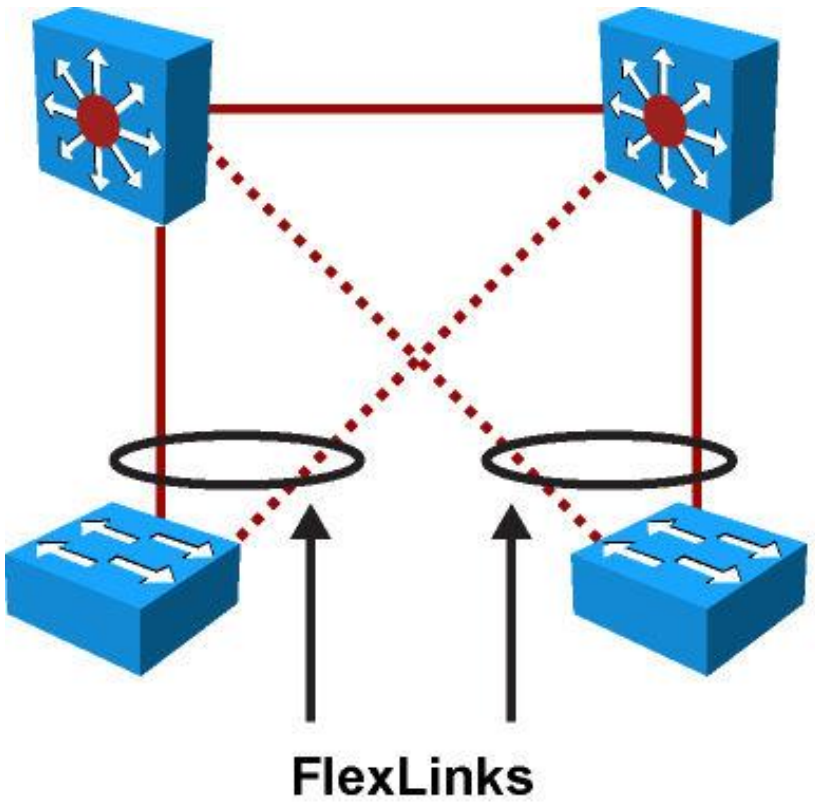
UDLD Configuration and Verification

```
Switch(config)# interface gigabitEthernet 5/1
Switch(config-if)# udld port aggressive
Switch# show udld gigabitEthernet 5/1
Interface Gi5/1
---
Port enable administrative configuration setting: Enabled / in aggressive mode
Port enable operational state: Enabled / in aggressive mode
Current bidirectional state: Bidirectional
Current operational state: Advertisement - Single neighbor detected
Message interval: 15
Time out interval: 5
Entry 1
---
Expiration time: 38
Device ID: 1
Current neighbor state: Bidirectional
Device name: FOX06310RW1
Port ID: Gi1/1
Neighbor echo 1 device: FOX0627A001
Neighbor echo 1 port: Gi5/1
Message interval: 15
Time out interval: 5
CDP Device name: SwitchB
```

Loop Guard versus Aggressive Mode UDLD

	Loop Guard	Aggressive Mode UDLD
Configuration	Per port	Per port
Action granularity	Per VLAN	Per port
Auto-recovery	Yes	Yes, with err-disable timeout feature
Protection against STP failures caused by unidirectional links	Yes, when enabled on all root ports and alternate ports in redundant topology	Yes, when enabled on all links in redundant topology
Protection against STP failures caused by problem in software in designated bridge not sending BPDUs	Yes	No
Protection against miswiring	No	Yes

Flex Links



- Flex Links is a Layer 2 availability feature that provides an alternative solution to STP and allows users to turn off STP and still provide basic link redundancy.
- Flex Links can coexist with spanning tree on the distribution layer switches; however, the distribution layer switches are unaware of the Flex Links feature.
- Flex Links enables a convergence time of less than 50 milliseconds. In addition, this convergence time remains consistent regardless of the number of VLANs or MAC addresses configured on switch uplink ports.
- Flex Links is based on defining an active/standby link pair on a common access switch. Flex Links are a pair of Layer 2 interfaces, either switchports or port channels, that are configured to act as backup to other Layer 2 interfaces.

Flex Links Configuration Considerations

- A Flex Link is configured on one Layer 2 interface (the active link) by assigning another Layer 2 interface as the Flex Link or backup link. When one of the links is up and forwarding traffic, the other link is in standby mode, ready to begin forwarding traffic if the other link shuts down. At any given time, only one of the interfaces is in the link up state and forwarding traffic. If the primary link shuts down, the standby link starts forwarding traffic. When the active link comes back up, it goes into standby mode and does not forward traffic.
- Flex Links are supported only on Layer 2 ports and port channels, not on VLANs or on Layer 3 ports.
- Only one Flex Link backup link can be configured for any active link.
- An interface can belong to only one Flex Link pair. An interface can be a backup link for only one active link. An active link cannot belong to another Flex Link pair.
- STP is disabled on Flex Link ports. A Flex Link port does not participate in STP, even if the VLANs present on the port are configured for STP.

Flex Links Configuration and Verification

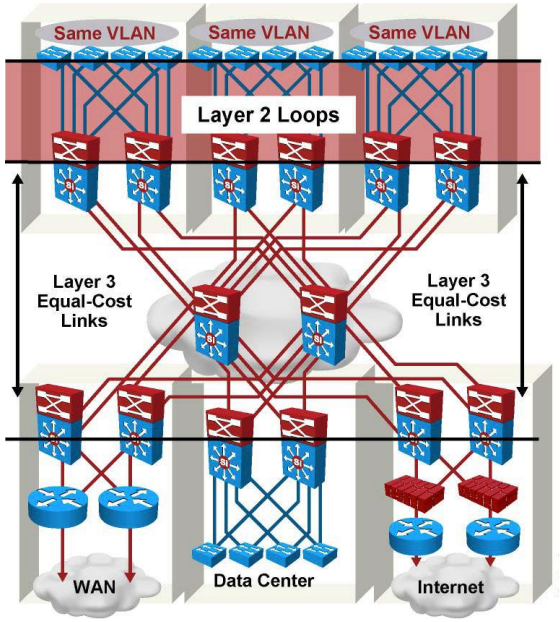
- FlexLinks are configured at the interface level with the command **`switchport backup interface`**.
- Here we configure an interface with a backup interface and verify the configuration.

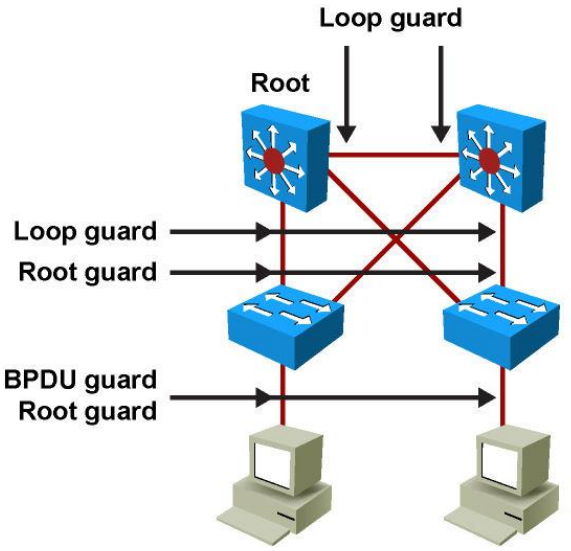
```
Switch(config)# interface fastethernet1/0/1
Switch(config-if)# switchport backup interface fastethernet1/0/2
Switch(config-if)# end
Switch# show interface switchport backup
Switch Backup Interface Pairs:
Active Interface     Backup Interface     State
-----------------    -----------------    --------------------
FastEthernet1/0/1    FastEthernet1/0/2    Active Up/Backup Standby
```

STP Best Practices and Troubleshooting

Switching Design Best Practices

- Use Layer 3 connectivity at the distribution and core layers.
- Use PVRST+ or MST. Do not disable STP at the access layer. Isolate different STP domains in a multivendor environment.
- Use Loop Guard on Layer 2 ports between distribution switches and on uplink ports from access to distribution switches.
- Use Root Guard on distribution switches facing access switches.
- Use Port security, PortFast, BPDU Guard, and Root Guard on access switch ports facing end stations.
- Use aggressive mode UDLD on ports linking switches.

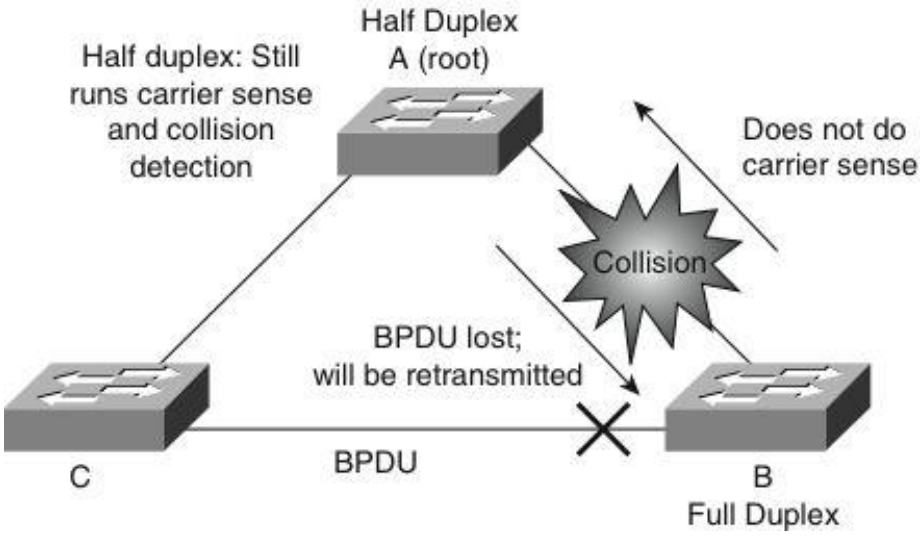




Potential STP Problems

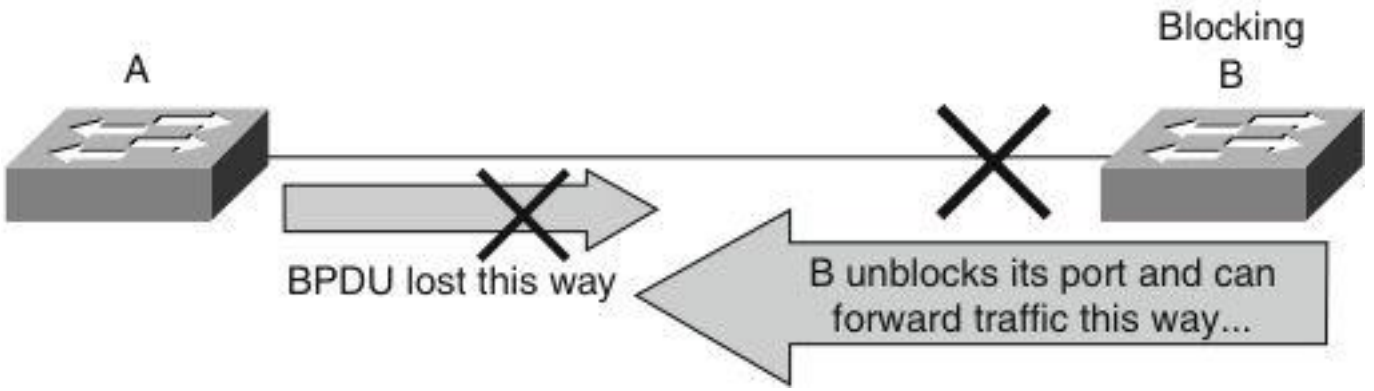
- Duplex mismatch
- Unidirectional link failure
- Frame corruption
- Resource errors
- PortFast configuration error

Duplex Mismatch



- Point-to-point link.
- One side of the link is manually configured as full duplex.
- Other side is using the default configuration for auto-negotiation.

Unidirectional Link Failure



- Frequent cause of bridge loops.
- Undetected failure on a fiber link or a problem with a transceiver.

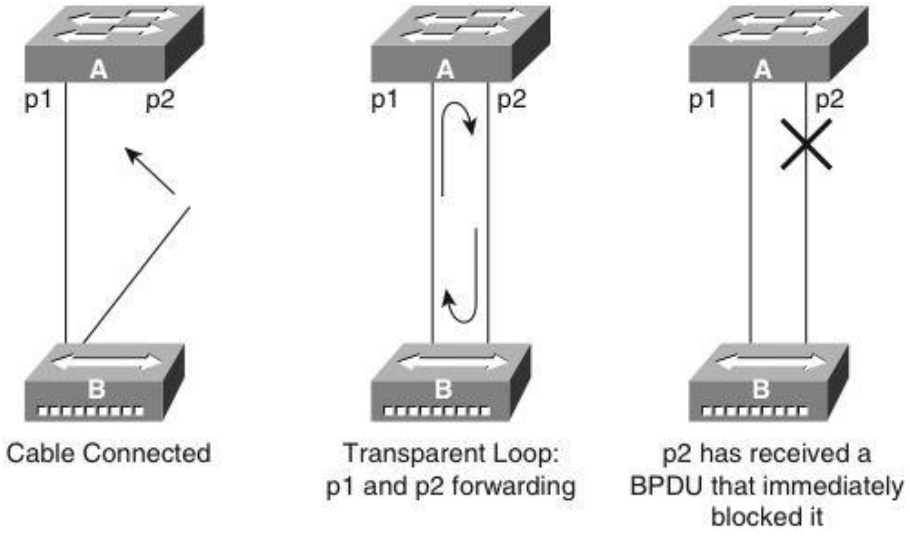
Frame Corruption

- If an interface is experiencing a high rate of physical errors, the result may be lost BPDUs, which may lead to an interface in the blocking state moving to the forwarding state.
- Uncommon scenario due to conservative default STP parameters.
- Frame corruption is generally a result of a duplex mismatch, bad cable, or incorrect cable length.

Resource Errors

- STP is performed by the CPU (software-based). This means that if the CPU of the bridge is over-utilized for any reason, it might lack the resources to send out BPDUs.
- STP is generally not a processor-intensive application and has priority over other processes; therefore, a resource problem is unlikely to arise.
- Exercise caution when multiple VLANs in PVST+ or PVRST+ mode exist. Consult the product documentation for the recommended number of VLANs and STP instances on any specific switch to avoid exhausting resources.

PortFast Configuration Error



- Switch A has Port p1 in the forwarding state and Port p2 configured for PortFast. Device B is a hub. Port p2 goes to forwarding and creates a loop between p1 and p2 as soon as the second cable plugs in to Switch A. The loop ceases as soon as p1 or p2 receives a BPDU that transitions one of these two ports into blocking mode.
- The problem with this type of transient loop condition is that if the looping traffic is intensive, the bridge might have trouble successfully sending the BPDU that stops the loop. BPDU guard prevents this type of event from occurring.

Troubleshooting Methodology

- Troubleshooting STP issues can be difficult if logical troubleshooting procedures are not deployed in advance. Occasionally, rebooting of the switches might resolve the problem temporarily, but without determining the underlying cause of the problem, the problem is likely to return. The following steps provide a general overview of a methodology for troubleshooting STP:
- Step 1. Develop a plan.
- Step 2. Isolate the cause and correct an STP problem.
- Step 3. Document findings.

Chapter 2 Summary (1)

- Spanning Tree Protocol is a fundamental protocol to prevent Layer 2 loops and at the same time provide redundancy in the network. This chapter covered the basic operation and configuration of RSTP and MST. Enhancements now enable STP to converge more quickly and run more efficiently.
  - RSTP provides faster convergence than 802.1D when topology changes occur.
  - RSTP enables several additional port roles to increase the overall mechanism's efficiency.
  - **`show spanning-tree`** is the main family of commands used to verify RSTP operations.
  - MST reduces the encumbrance of PVRST+ by allowing a single instance of spanning tree to run for multiple VLANs.

Chapter 2 Summary (2)

- The Cisco STP enhancements provide robustness and resiliency to the protocol. These enhancements add availability to the multilayer switched network. These enhancements not only isolate bridging loops but also prevent bridging loops from occurring. To protect STP operations, several features are available that control the way BPDUs are sent and received:
  - BPDU guard protects the operation of STP on PortFast-configured ports.
  - BPDU filtering prevents BPDUs from being sent and ignores received BPDUs while leaving the port in forwarding state.
  - Root guard prevents root switch being elected via BPDUs received on a root-guard configured port.
  - Loop guard detects and disables an interface with Layer 2 unidirectional connectivity, protecting the network from anomalous STP conditions.
  - UDLD detects and disables an interface with unidirectional connectivity, protecting the network from anomalous STP conditions.
  - In most implementations, the STP toolkit should be used in combination with additional features such as Flex Links.

Chapter 2 Labs

- **SW-LAB-1**
  - **Rapid Spanning Tree Protocol**
  - **Multiple Spanning Tree Protocol**

Q&A