

# **11W NET3011**

## **CCNP SWITCH – Chapter 3**

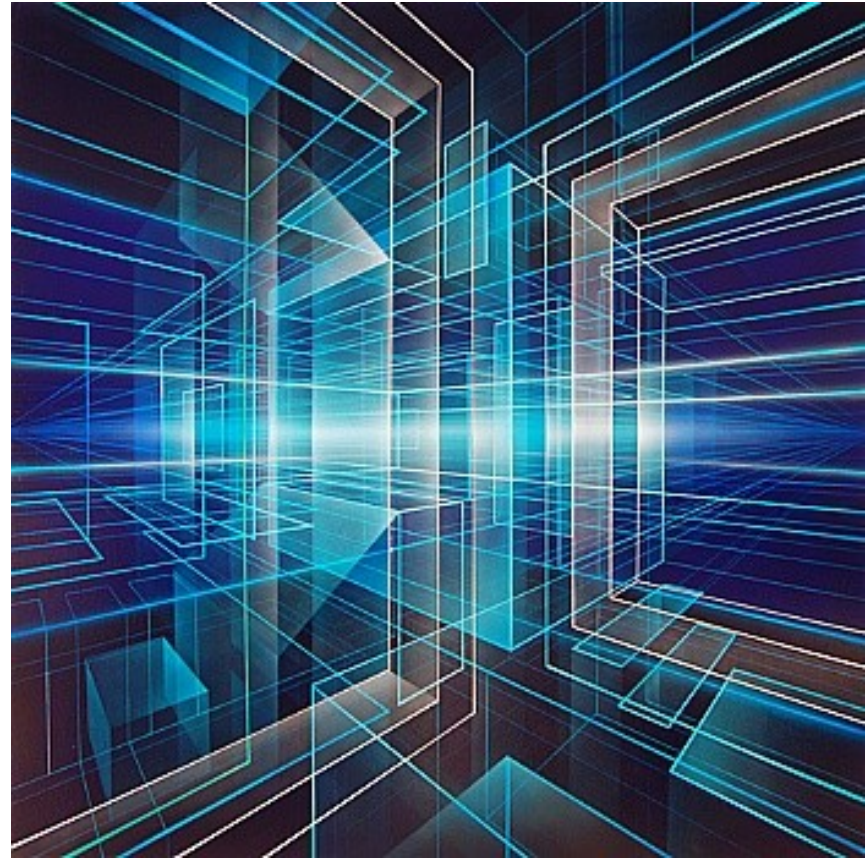
### **STP – Part 2**

**David Bray**

**brayd@algonquincollege.com**

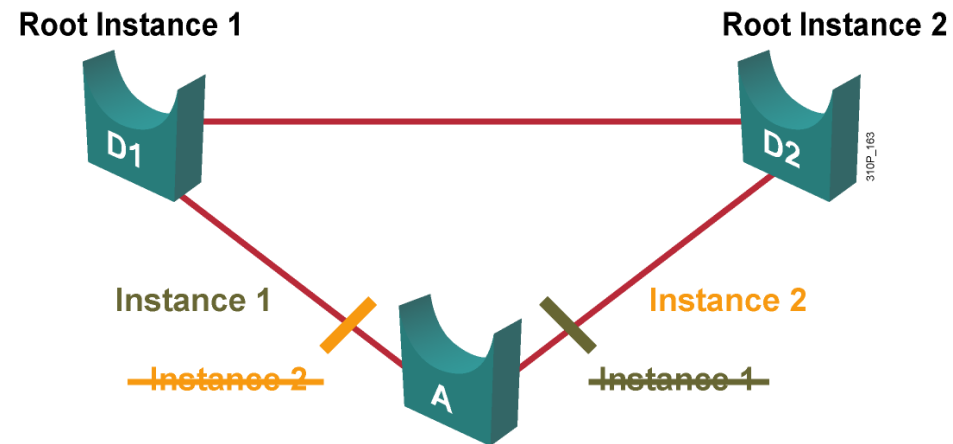
with contributions obtained from Rick Graziani & Cisco

# Multiple STP



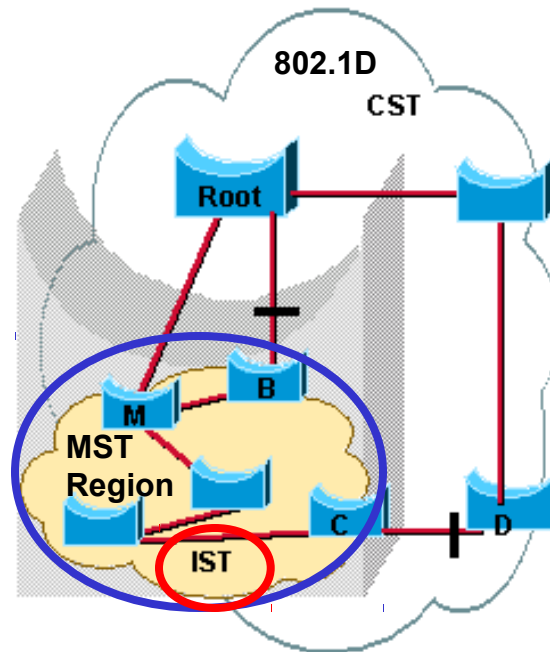
# Multiple Spanning Tree Protocol – 802.1s

Instance 1 maps to VLANs 1–500  
Instance 2 maps to VLANs 501–1000



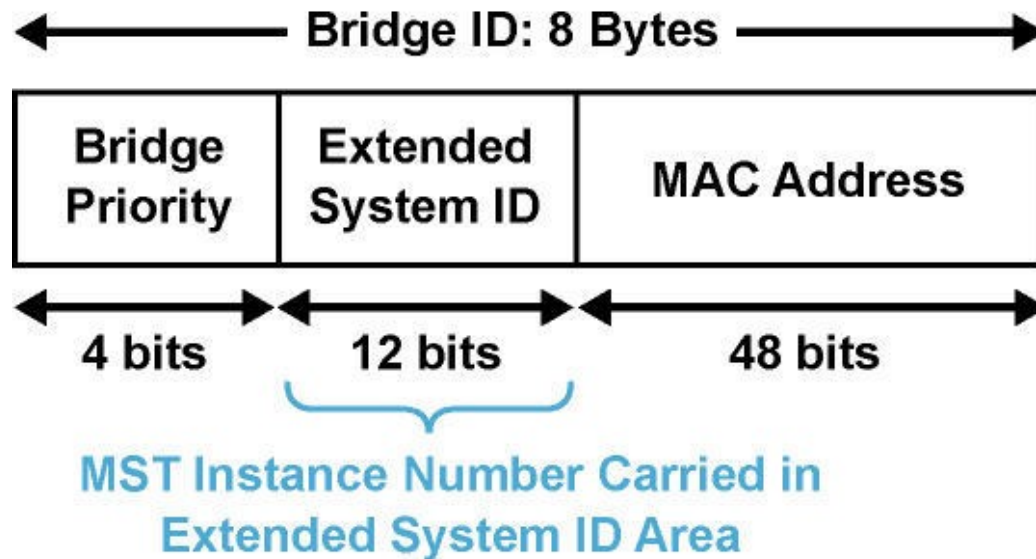
- **Multiple Spanning Tree (MST)** extends the IEEE 802.1w RST algorithm to multiple spanning trees.
- The main purpose of MST is to:
  - Reduce the total number of spanning-tree instances to match the physical topology of the network
  - Thus reduce the CPU cycles of a switch.
- Allows the network administrator to configure the exact number of instances.
- **PVST+** runs a **single instance of STP for each VLAN and does not take into consideration the physical topology.**
  - May have **1,000 VLANs** but only **2 different topologies** (2 different Root Bridges).
  - PVST+ will still create **1,000 instances of STP**
- **MST**, on the other hand, uses a **minimum number of STP instances** to match the number of physical topologies present.
  - May have **1,000 VLANs** but only **2 different topologies** (2 different Root Bridges).
  - MST will let you specify only **2 instances of STP.**

# MST Regions



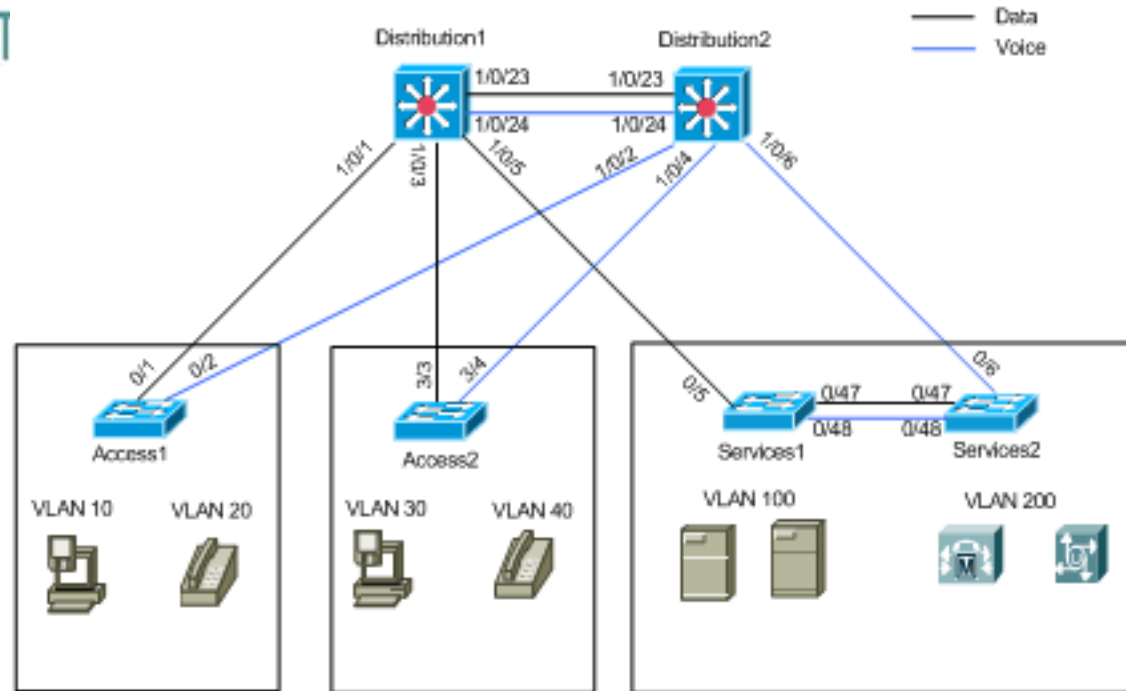
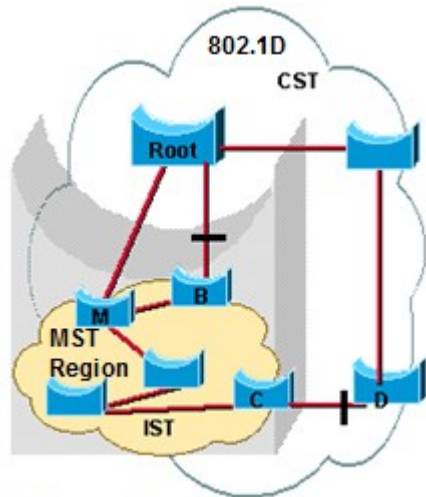
- An MST Region is a group of switches placed under a common administration (like an AS).
- In most networks a single MST region is sufficient.
  - A single MST Region can handle 15 user STP instances (topologies).
- Within a region, all switches must run the instance of MST as defined by a common:
  - MST configuration **name** (32 characters)
  - MST configuration **revision number** ( 0 to 65,535)
  - MST instance-to-VLAN **mapping table** (4,096 entries)
- MST was designed to coexist with CSTP (802.1D) & RSTP (802.1w).
- A special instance called the **IST (Internal Spanning Tree - instance 0)**, exists to ensure a loop-free topology between devices inside the MST Region and those outside it.
- IST presents the entire MST region as a single virtual switch (bridge) to all external devices (those not configured within MST).

# MST Use of Extended System ID



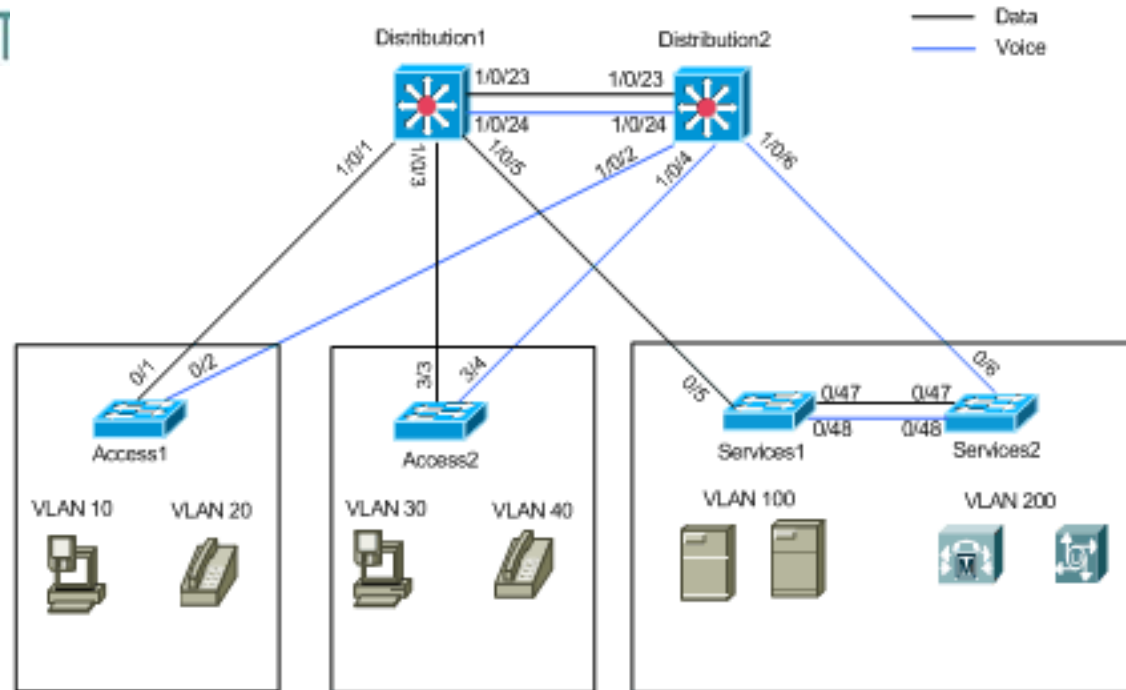
- MST carries the instance number in the 12-bit Extended System ID field of the Bridge ID.

# Quick Example



- Remember, the whole idea of MST is to map multiple VLANs to a smaller number of STP instances.
  - Cisco supports a **maximum of 16 MST Instances (MSTIs) in a region.**
  - The **IST uses MST 0** leaving 1 through 15 available for use.
- The **Distribution1** switch is the primary root bridge for the data VLANs 10, 30, and 100
  - Secondary root bridge for the voice VLANs 20, 40, and 200.
- The **Distribution2** switch the primary root bridge for the voice VLANs 20, 40, and 200
  - Secondary root bridge for the data VLANs 10, 30, and 100.
- Distribution1** is chosen as **CIST regional root.**
  - It means that Distribution1 is the root for IST0.

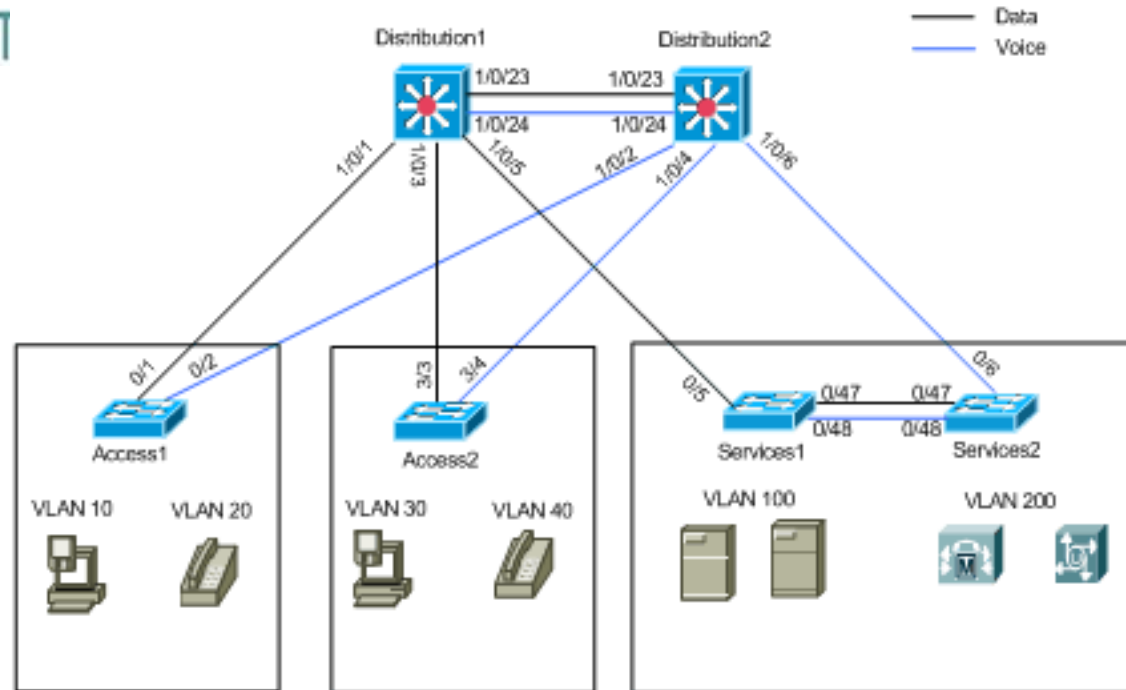
# MST – Dist1



```

Distribution1(config)# spanning-tree mode mst          Enables MST
Distribution1(config)# spanning-tree mst configuration
Distribution1(config-mst)# name region1              Configure Region
                                                         and MST instances
Distribution1(config-mst)# revision 10
Distribution1(config-mst)# instance 1 vlan 10, 30, 100
Distribution1(config-mst)# instance 2 vlan 20, 40, 200
Distribution1(config-mst)# exit                      Configure Root Bridge
Distribution1(config)# spanning-tree mst 0-1 root primary
Distribution1(config)# spanning-tree mst 2 root secondary
  
```

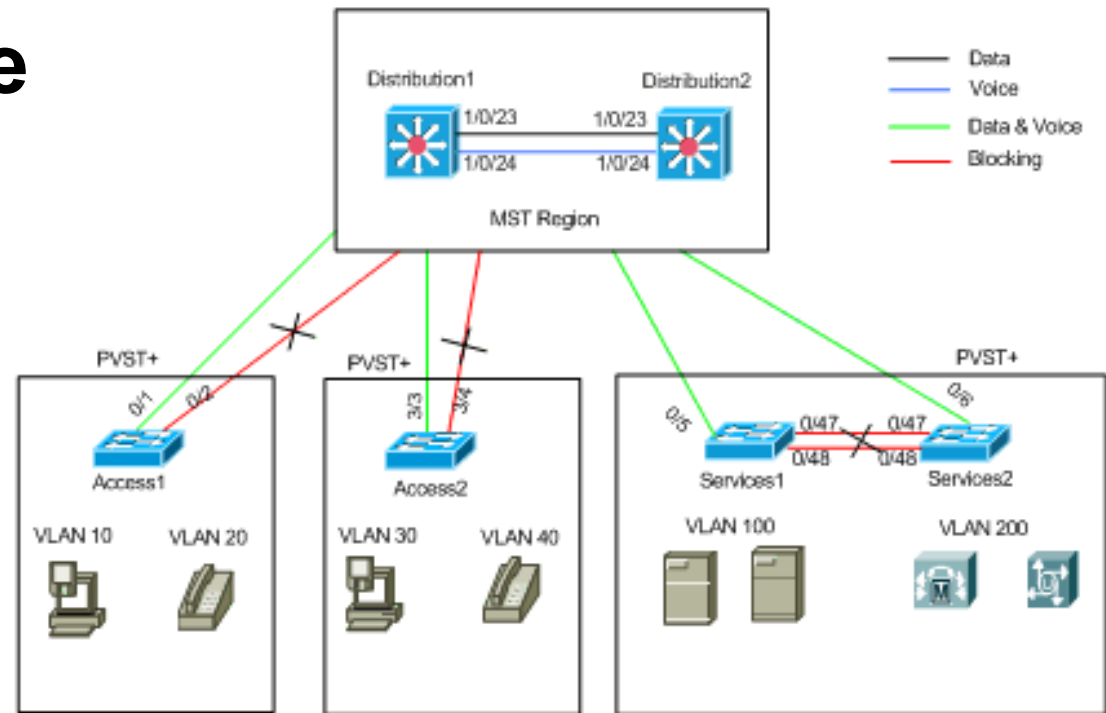
# MST – Dist2



```

Distribution2 (config) # spanning-tree mode mst           Enables MST
Distribution2 (config) # spanning-tree mst configuration
Distribution2 (config-mst) # name region1               Configure Region
                                                                and MST instances
Distribution2 (config-mst) # revision 10
Distribution2 (config-mst) # instance 1 vlan 10, 30, 100
Distribution2 (config-mst) # instance 2 vlan 20, 40, 200
Distribution2 (config-mst) # exit                       Configure Root Bridge
Distribution2 (config) # spanning-tree mst 2 root primary
Distribution2 (config) # spanning-tree mst 0-1 root secondary
  
```

# MST – Reference



- For complete configurations go to:
- **Configuration example to migrate Spanning Tree from PVST+ to MST**
- [http://www.cisco.com/en/US/products/hw/switches/ps708/products\\_configuration\\_example09186a00807b075f.shtml](http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_example09186a00807b075f.shtml)

# MST

```
Switch# show spanning-tree
```

```
MST00
```

```
Spanning tree enabled protocol mstp
```

```
Root ID      Priority    24577
              Address    0001.C945.A573
              Cost      4
              Port      26(GigabitEthernet1/2)
              Hello Time 2 sec    Max Age 20 sec    Forward Delay 15 sec

Bridge ID    Priority    32769   (priority 32768 sys-id-ext 1)
              Address    0003.E461.46EC
              Hello Time 2 sec    Max Age 20 sec    Forward Delay 15 sec
              Aging Time 20
```

# MST Configuration Commands

- Enable MST on the switch.

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

- Enter MST configuration submode.

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

- Display current MST configuration – changes are not committed until you exit MST submode.

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

- Name the MST region – each switch can only participate in a single region.

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

- Explicitly set the 16-bit MST revision number. Unlike VTP, it is not automatically incremented when you commit a new MST configuration.

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

# MST Configuration Commands (cont'd)

- Map VLANs to an MST instance.

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

- Display new MST configuration about to be applied. Helpful in verifying the modified configuration prior to commital.

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

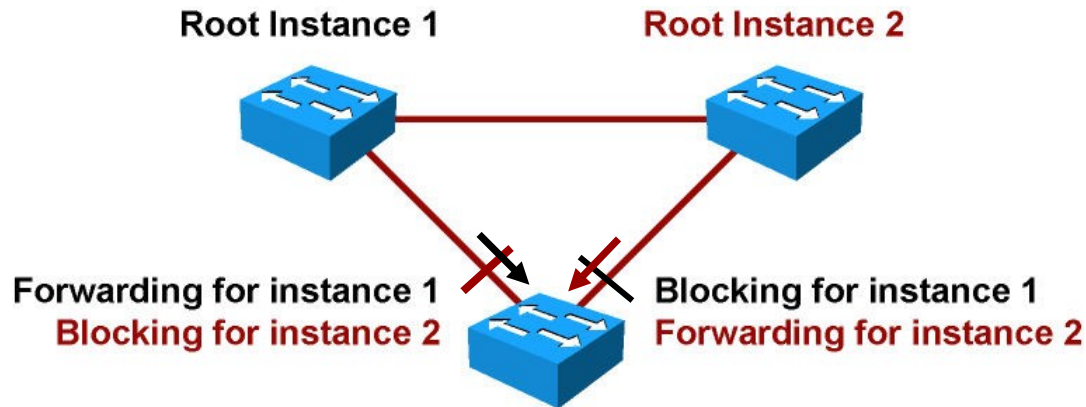
- Exits MST configuration submode and applies latest configuration changes.

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

- Influences root bridge for the specified MST instance. As in regular PVST+, this is achieved by setting 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
```

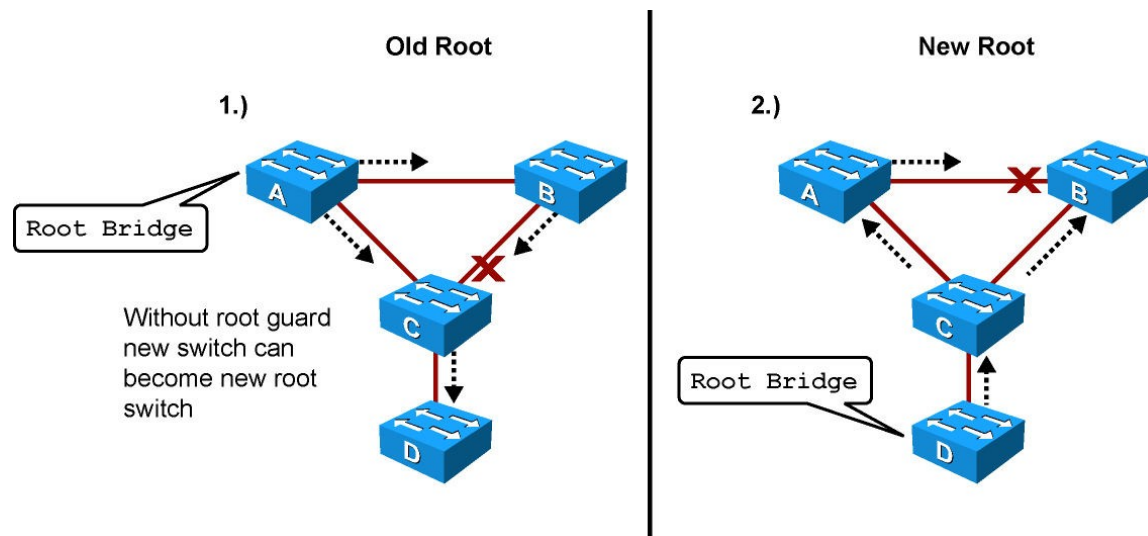


# More STP Enhancements

# Root Guard

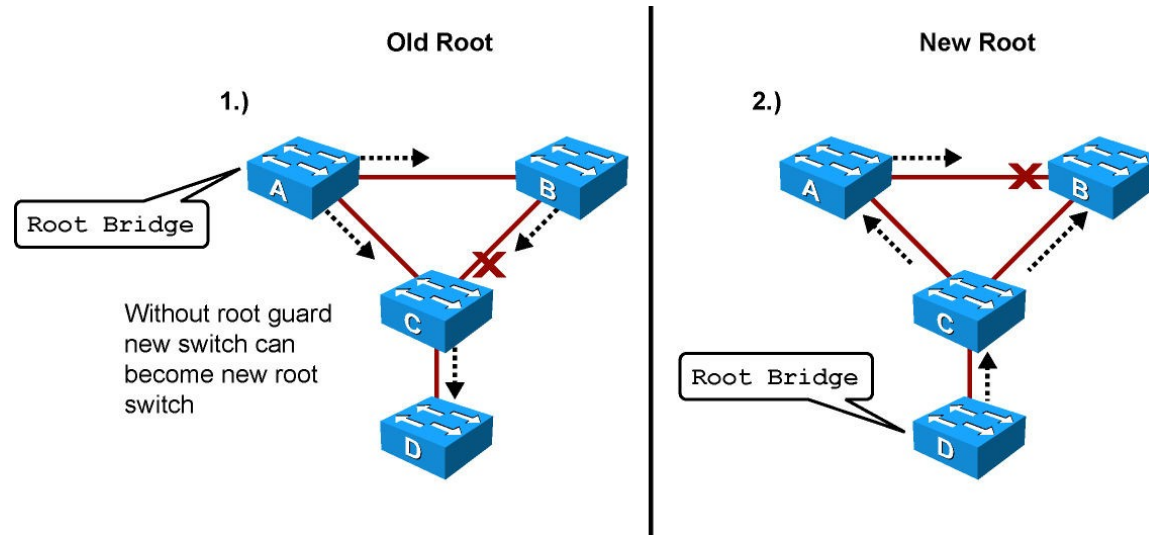
- Root guard is useful in avoiding inefficiencies or Layer 2 loops during network anomalies.
  - The Root guard feature is enabled on a per-port basis and **forces that port to become a DP**, thereby preventing any connected downstream switches from becoming the root bridge.
- If a **root guard**-enabled port receives a superior BPDU it is immediately moved to a *root-inconsistent* STP state.
  - Effectively, this is equivalent to Listening state and therefore prevents traffic forwarding on that port.
- As a result, this feature enforces the position of the root bridge because it doesn't allow any downstream device to join the active topology and declare itself as the new root.
  - “No downstream device can join our network as the root – at least NOT through me!!”

# Root Guard Motivation



- Suppose switches A and B are connected by a gigabit link in the core of the network, with switch A being 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 an undesirable outcome.

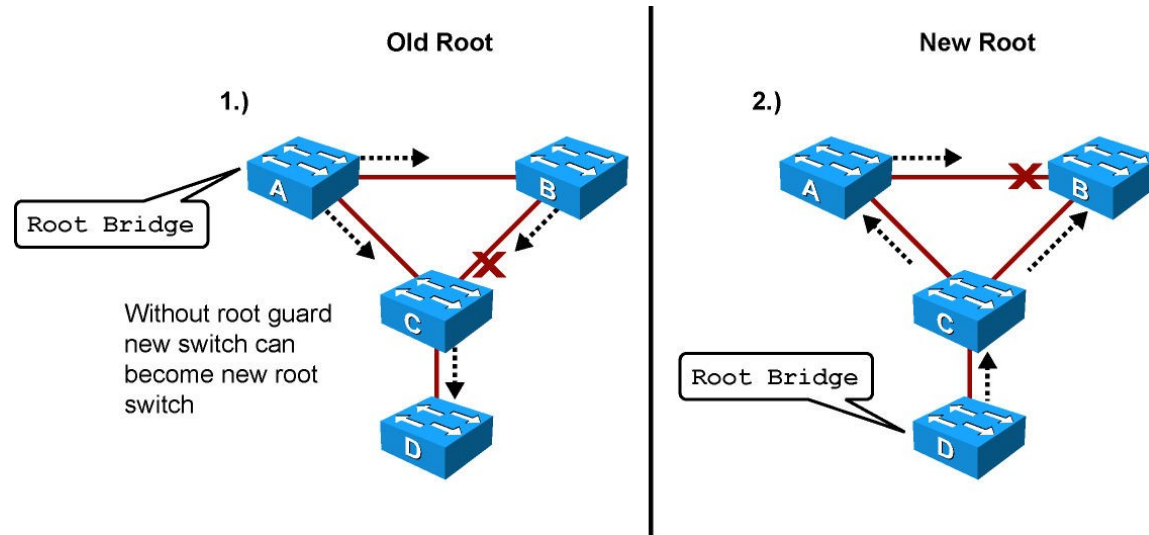
# Root Guard Operation



- With the root guard feature enabled on the port leading to Switch D, it cannot become an STP root port and so, prevents D from becoming the root bridge.
- The message similar to the following will be logged 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 design recommendation is to enable root guard on all access ports so that a root bridge is not established through these ports.
- In our example, Switch C blocks the port connecting to Switch D when it receives a superior BPDU, by transitioning the port to *root-inconsistent* state. No traffic passes through a port in this state.
- When Switch D stops sending superior BPDUs, the port automatically unblocks, and proceeds through regular STP transitions of Listening, Learning, and eventually to the Forwarding state. Recovery is fully automatic with no administrative intervention 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
```

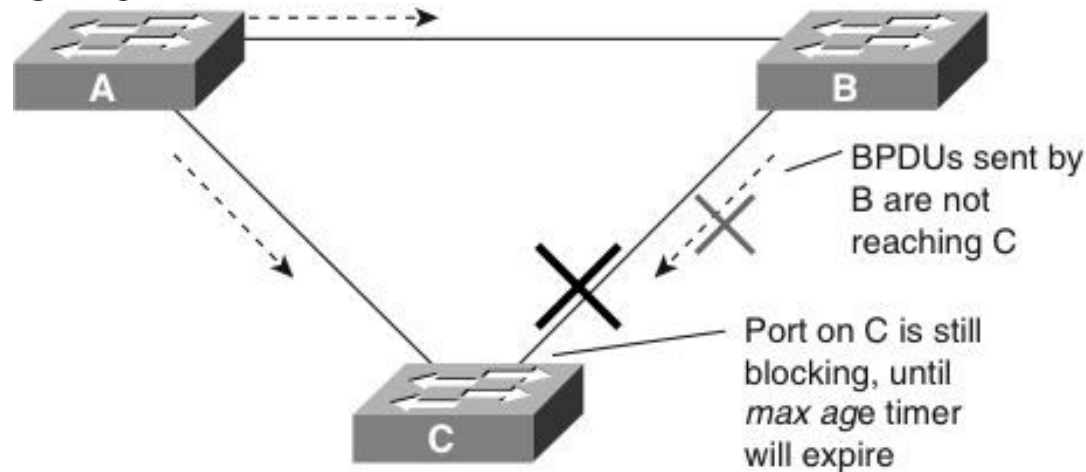
Name	Interface	Inconsistency
-----	-----	-----
VLAN0001	FastEthernet3/1	Port Type Inconsistent
VLAN0001	FastEthernet3/2	Port Type Inconsistent
VLAN1002	FastEthernet3/1	Port Type Inconsistent
VLAN1002	FastEthernet3/2	Port Type Inconsistent

Number of inconsistent ports (segments) in the system :4

# Unidirectional Links

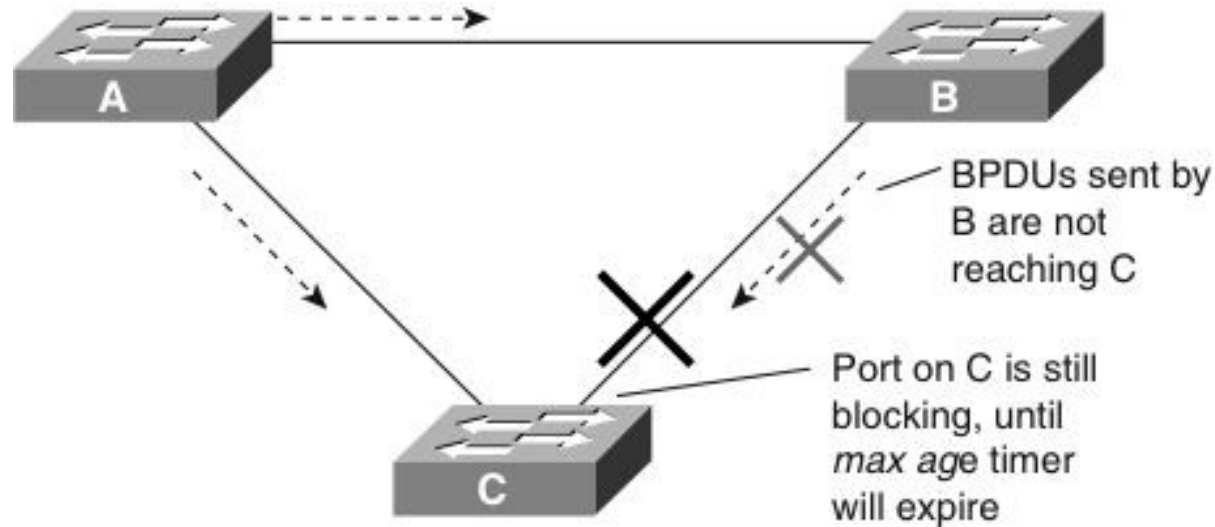
- Loss of BPDUs can sometimes be caused by unidirectional links – those that can send frames, but cannot receive them.
- Such a situation is typically due to a problem at the Physical Layer, with causes like:
  - cabling/connector failure, for example on a fibre link where one direction of the full-duplex pair has been compromised
  - hardware failure (e.g. transceiver failure)
- The absence of BPDUs will eventually cause the port to mistakenly transition to a forwarding state, causing a bridging loop, possibly bringing down the network.
- The greatest risk occurs with trunk links.
- Two features can guard against such an outcome:
  - Loop Guard
  - Uni-Directional Link Detection (UDLD)

# Loop Guard



- 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.
- 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.
```

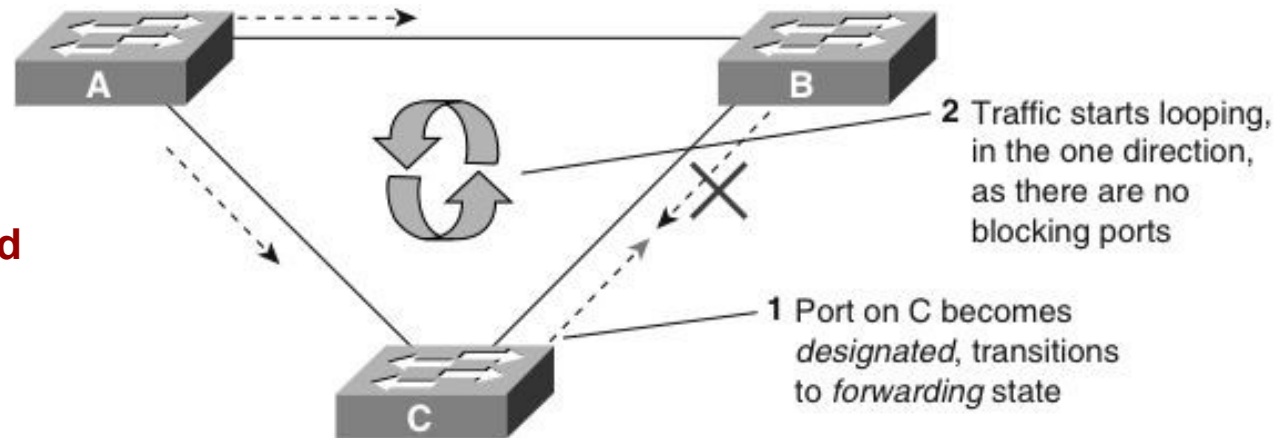
```
Moved to loop-inconsistent state.
```

- After recovery, the switch logs the following message:

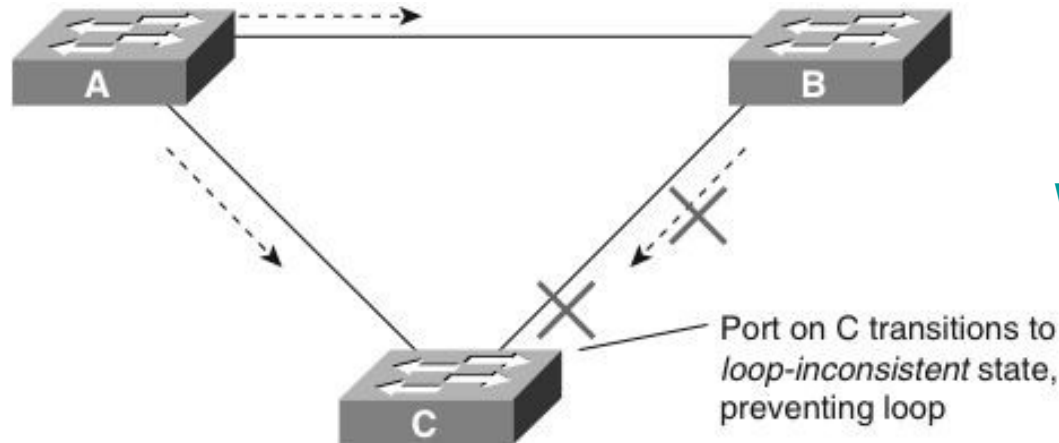
```
SPANTREE-2-LOOPGUARDUNBLOCK: port 3/2 restored in vlan 3.
```

# Loop Guard Operation

## Without Loop Guard

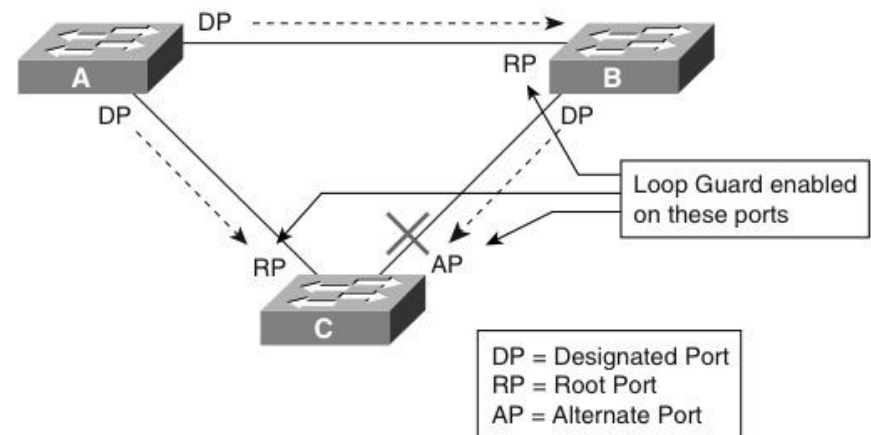


## With Loop Guard



# Loop Guard Configuration Considerations

- Loop Guard is disabled by default on Cisco switches.
- Although Loop Guard is configured on a per-port basis, on a trunk port, the feature blocks inconsistent ports on a per-VLAN basis; for example, if BPDUs are not received for only one particular VLAN, the switch blocks only that VLAN (that is, the port for only that VLAN, moves 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.



**Prevent upstream links from ever being mistaken for downstream pathways.**

# 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:

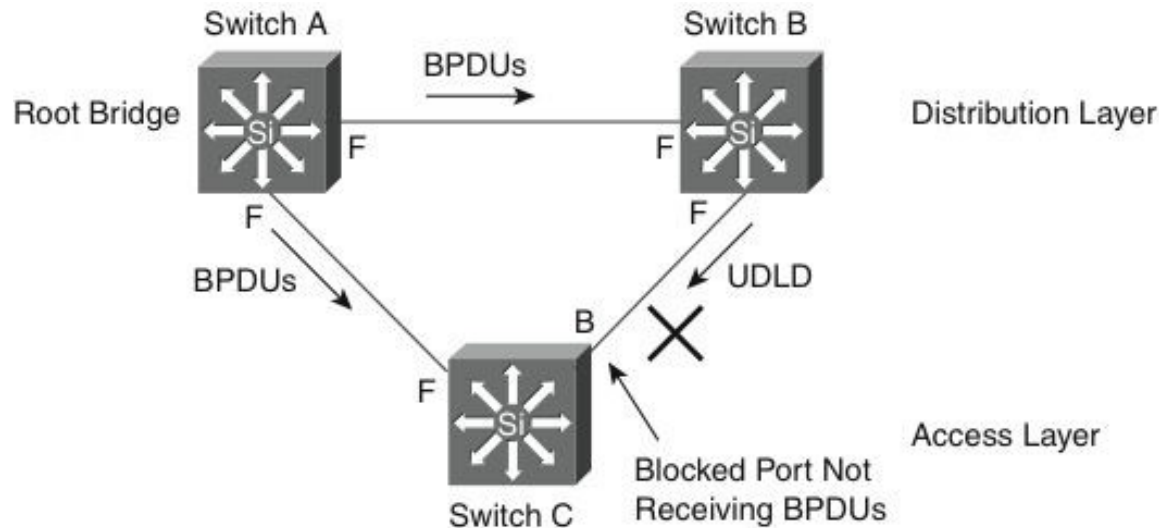
```
Switch(config) # spanning-tree loopguard default
```

# Verifying Loop Guard Configuration

- To verify Loop Guard status on an interface, issue the 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**.
  - `udld port` enables UDLD normal mode on an interface.
  - `udld port aggressive` UDLD aggressive mode on an interface.
- As expected, the interface command overrides any global setting.

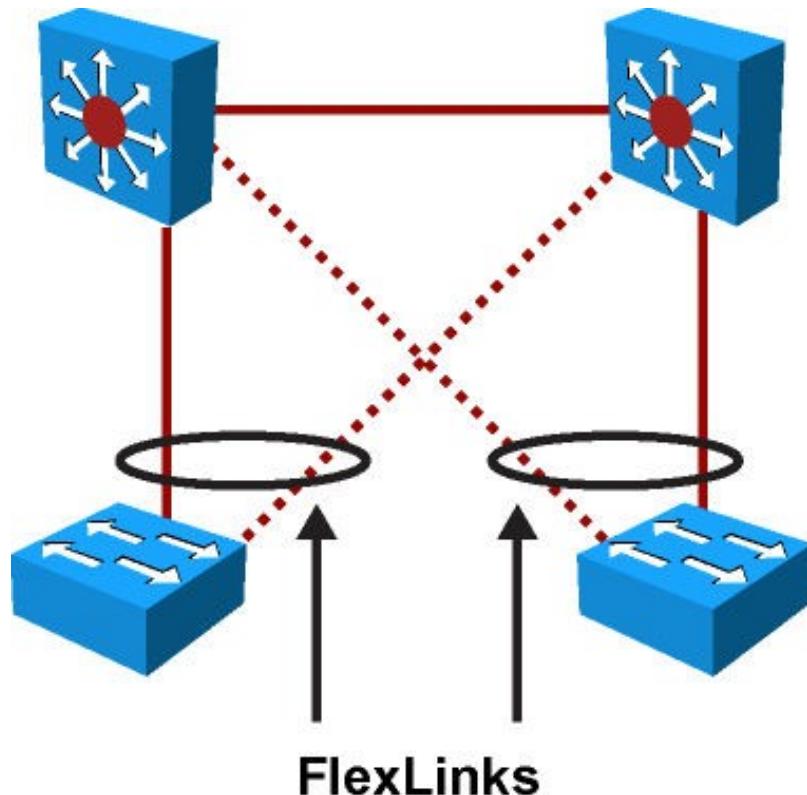
# UDLD Configuration and Verification

```
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.

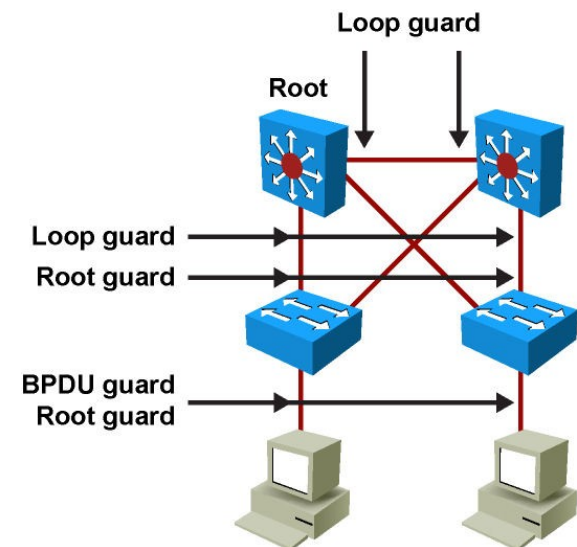
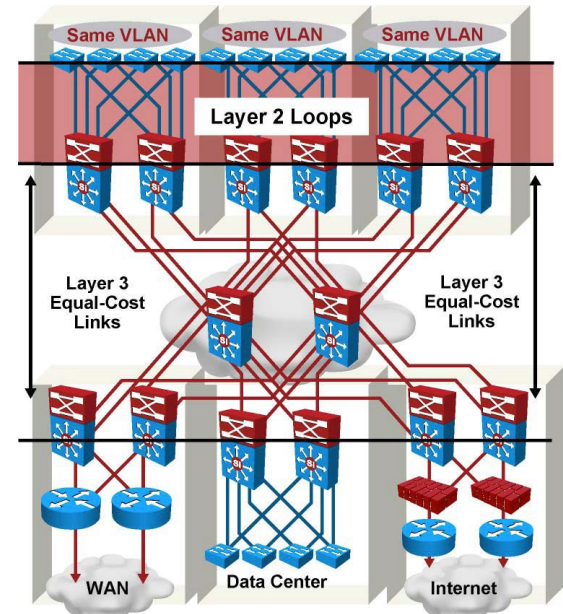
```
Switch(config)# interface fastethernet1/0/1
Switch(config-if)# switchport backup interface fastethernet1/0/2
Switch(config-if)# end
Switch# show interface switchport backup
Switch Backup Interface Pairs:
Active Interface          Backup Interface          State
-----
FastEthernet1/0/1        FastEthernet1/0/2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



# STP Wrap-Up

# Switching Design Best Practices

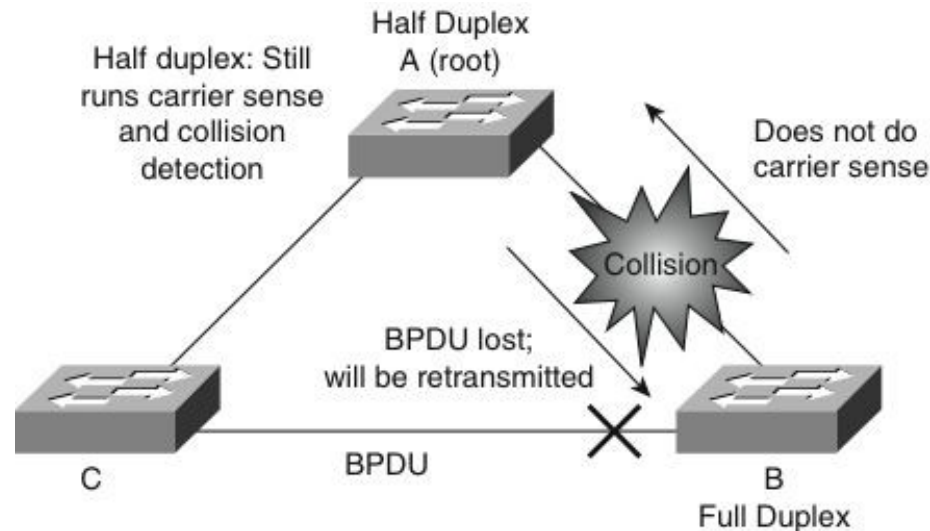
- Use Layer 3 connectivity at the distribution and core layers.
- Use RPVST+ or MST. Do not disable STP at the access layer. Isolate different STP domains in a multivendor environment.
- Use Loop Guard on Layer 2 ports between distribution switches and on uplink ports from access to distribution switches.
- 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.



# Other Potential STP Problems

- Duplex mismatch
- 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, resulting in HDX.
  - FDX side (B) sends frames at will
  - HDX side (A) is CSMA/CD so it's continually backing off for re-transmit
  - B fails to receive BPDUs from A => starts forwarding => bridging loop!

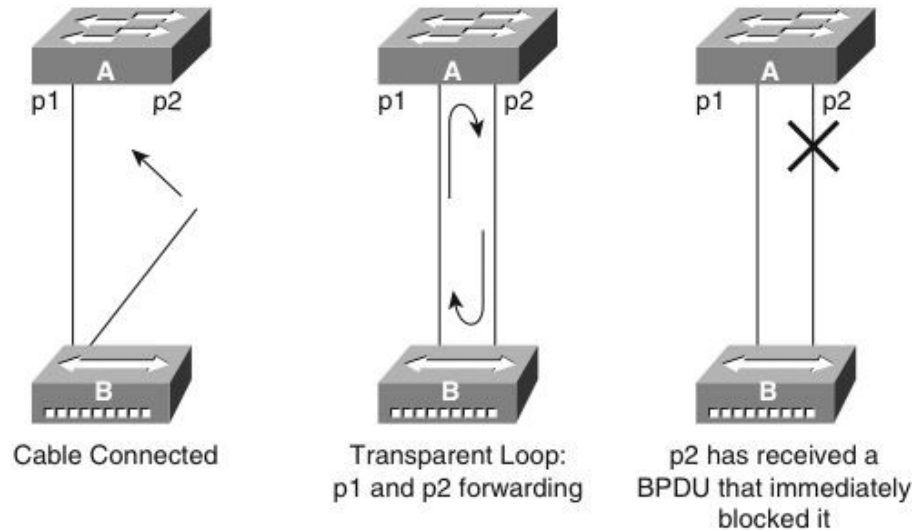
# Frame Corruption

- 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. However, it's still possible.
- 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 exist in PVST+ or RPVST+ mode. 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.
- **Problems that go away by themselves, come back by themselves!**
- 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.