Spanning Tree Protocol

SWITCH Module 3

Agenda

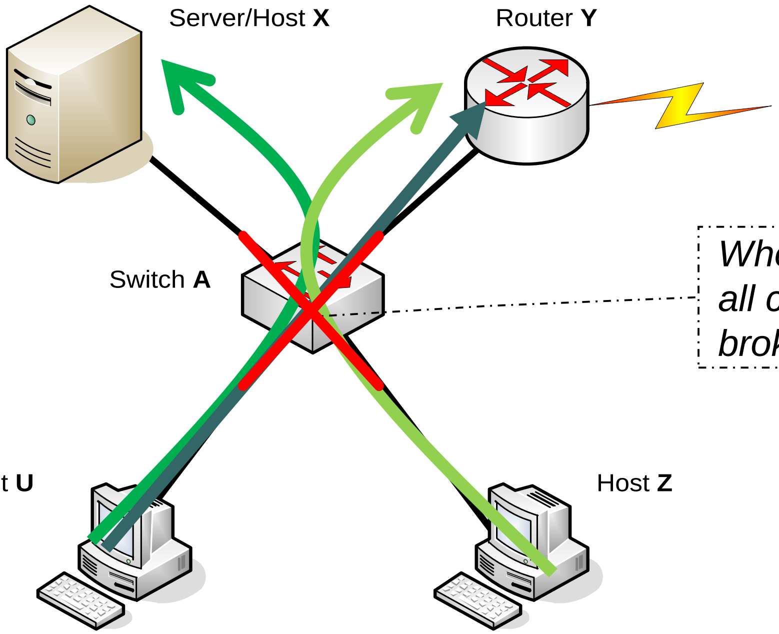
- STP Introduction
- Legacy STP
- RSTP
- MSTP
- Securing STP
- Flex Links

Some Facts about Switch Functionality

- Ethernet switches belongs to the family of so called transparent bridges
  - Connected hosts do not know about their existence
  - Switches do not alter passing frames
- Switches learn by reading source MAC addresses
- Relaying of frames is directed by destination MAC address and looking into corresponding MAC address table
  - Frames intended to known recipients – send it only through corresponding interface
  - Frames intended to unknown recipients – flood it through all interfaces except the one on which frame was received
- *This behavior causes some troubles in topologies with redundancy – either links or devices!*

Motivation for Redundancy

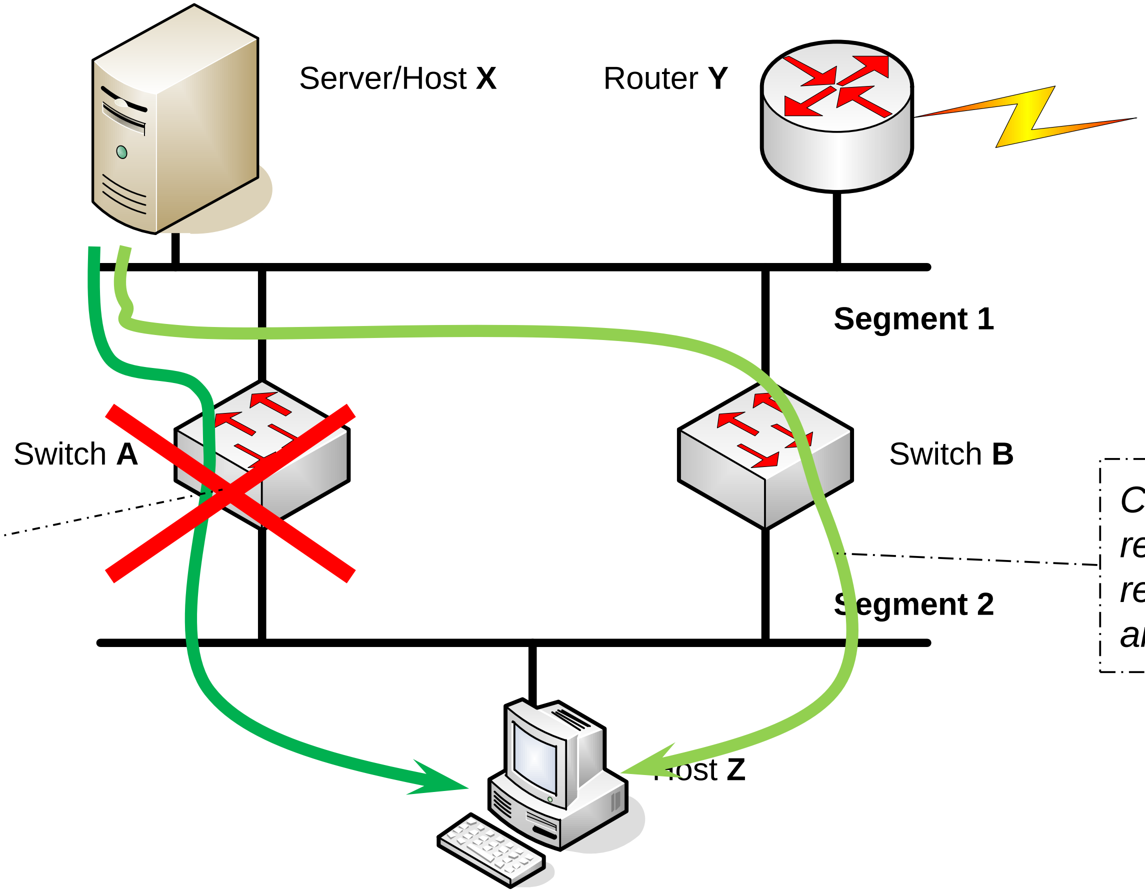
- Main goal of redundancy is to reduce results of network failures



When devices goes down all communication is broken!

- Solution to this problem is redundancy of devices and links

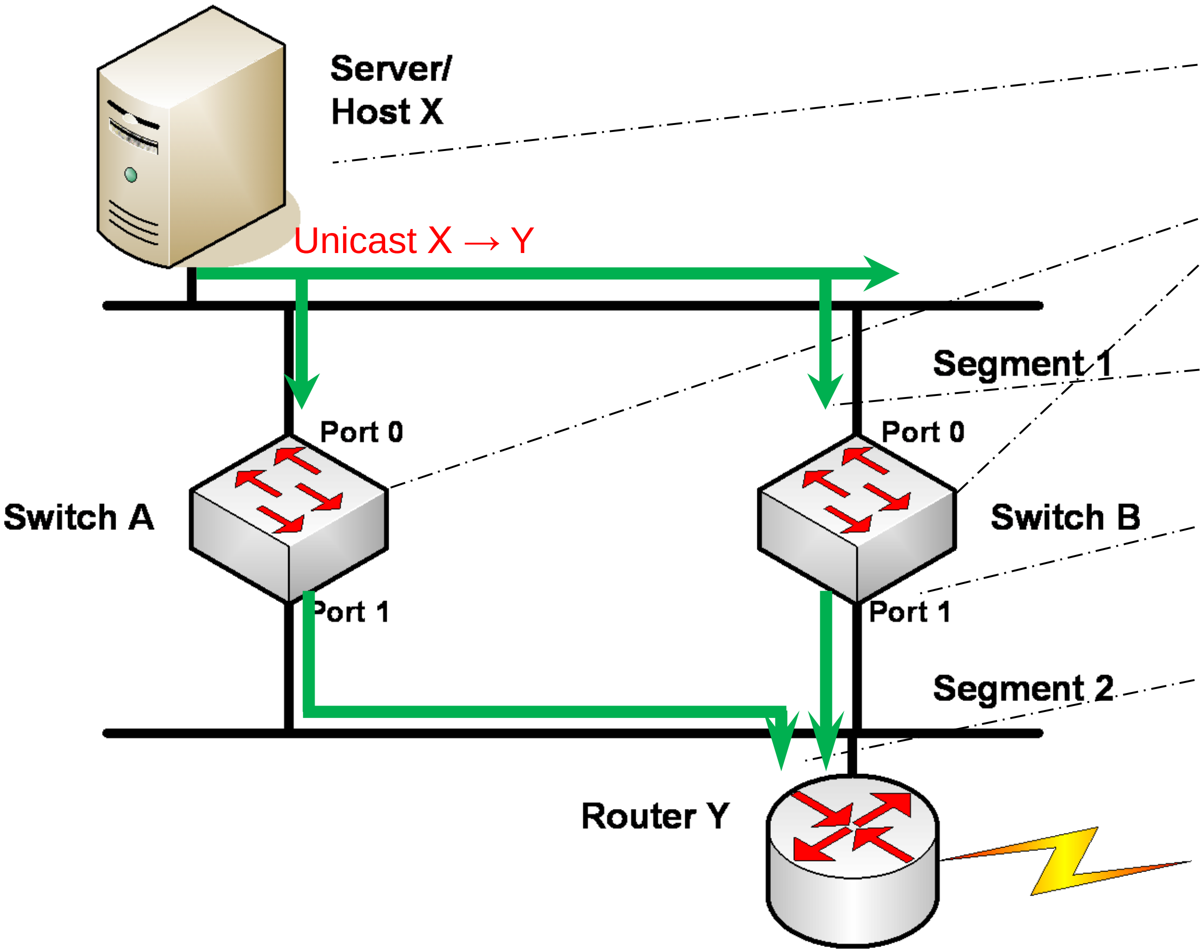
Simple Redundant Topology



Failure of Switch ***A***

Communication is restored via redundant link and device

Connectivity Troubles ①



ARP cache on host ***X*** *has record for router* ***Y***

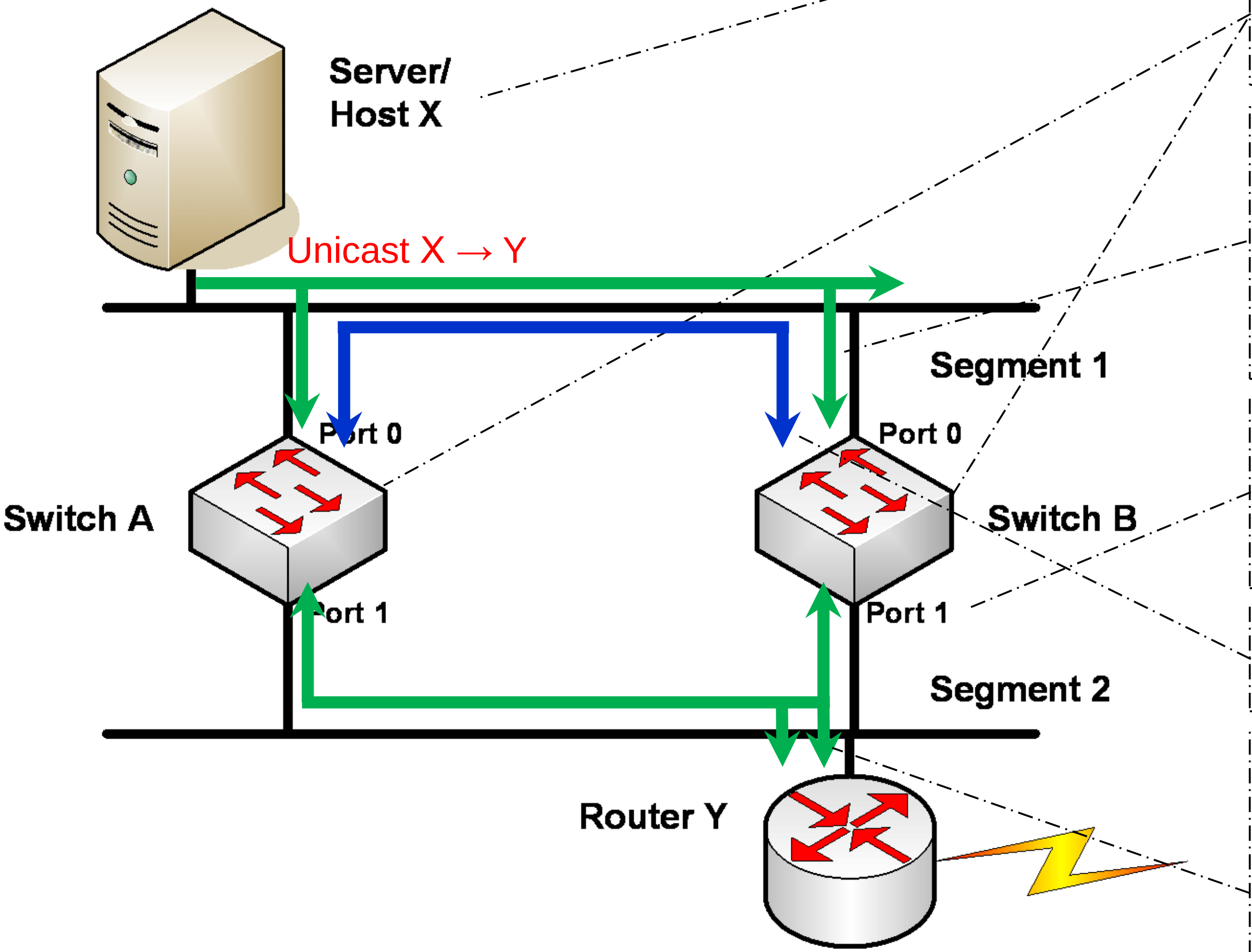
*Both switches have appropriate mapping of MAC-****X****/Port and MAC-****Y****/Port.*

Whenever ***X*** *sends unicast frame to* ***Y****, both switches receives this frame.*

Both switches send unicast frame towards router ***Y****.*

Problem ①***:***
Router ***Y*** *receives multiple copies of the same frame.*

Connectivity Troubles ②



*ARP cache on host **X** has record for router **Y**.*

*Both switches do not have appropriate mapping of MAC-**X**/Port and MAC-**Y**/Port.*

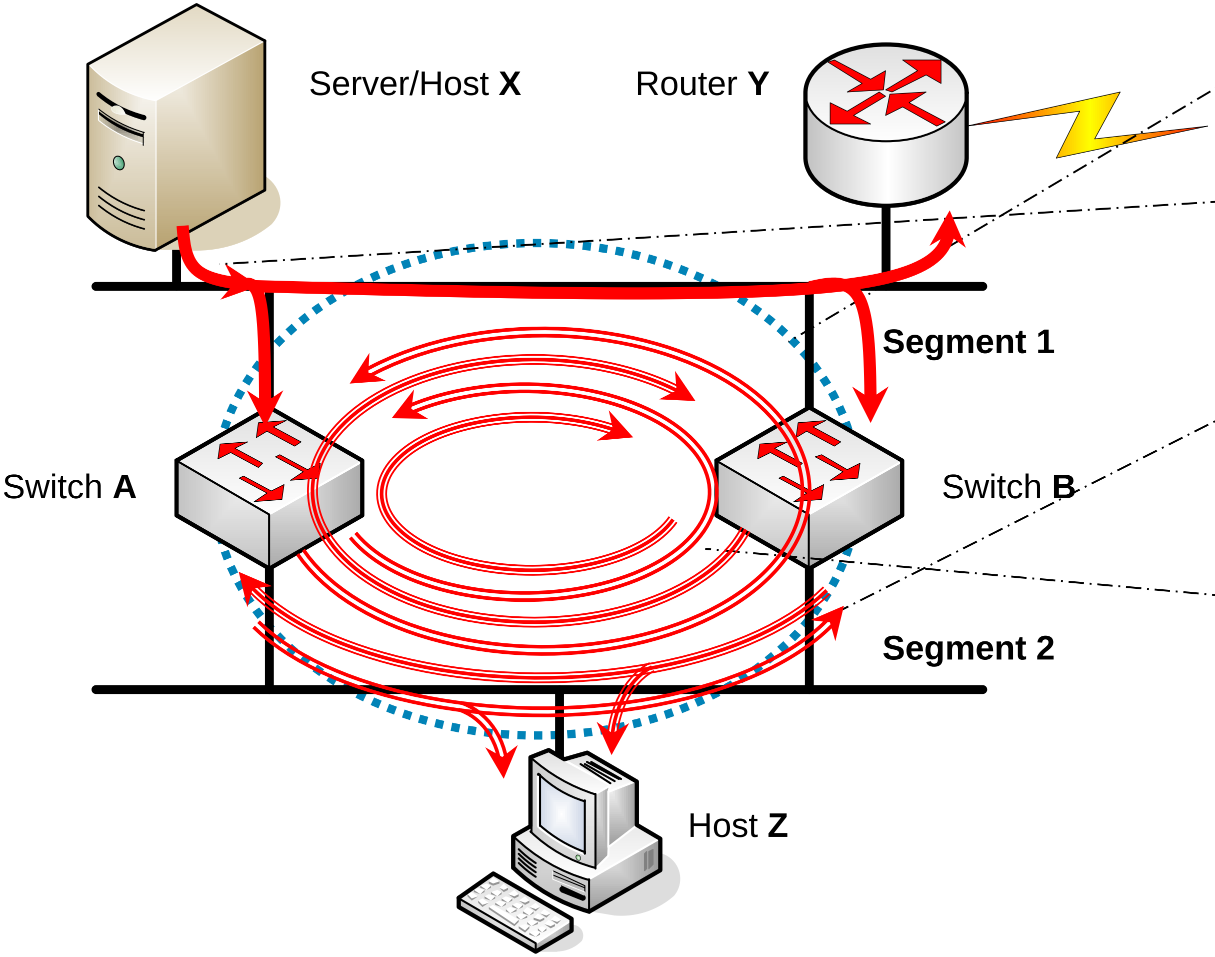
*Whenever **X** sends unicast frame to **Y**, both switches receives this frame. Because they do not know right destination interface for MAC-**Y**, they just flood frame.*

***A** receives frame from **B** and vice versa. Both switches create new mapping for MAC-**X**/Port on Segment 2.*

*MAC-**Y** is still unknown; hence, flooding situation repeats.*

Problem ②:
*Switches create bad mapping (**MAC address table instability**) for MAC-**X**/Port. Additionally to that frame is resend in infinite loop and router receives it multiple copies.*

Connectivity Troubles ③



Topology loop was created by adding 2nd switch.

Part of every Ethernet LAN communication is usually broadcast!

Received broadcast is flooded on every interface except of the incoming port. Same is done by 2nd switch.

Infinite resending of broadcast frames would occur because frames do not have any TTL.

Problem ③:
***Broadcast Storms** consume*
Broadcasty spotrebujú podstatnú časť prostriedkov siete i pripojených staníc.

Spanning Tree Protocol

- Physical loops in network are inevitable
  - They are only option to keep redundancy of devices and links functional
- *L2 loops elimination is what we want to accomplish in the reality!*
  - Forwarding loop caused by switch logic
- This problem is solved by STP-based protocols
  - There are variety of STP protocols but basic idea is same
  - STP creates spanning tree from graph point of view of network
  - Spanning tree is formed as shortest path tree from every switch to one reference point – root bridge
    - Because of total order and metric paths there could not be two or more equal paths
    - From switch’s point of view there will be always only one best path to root bridge

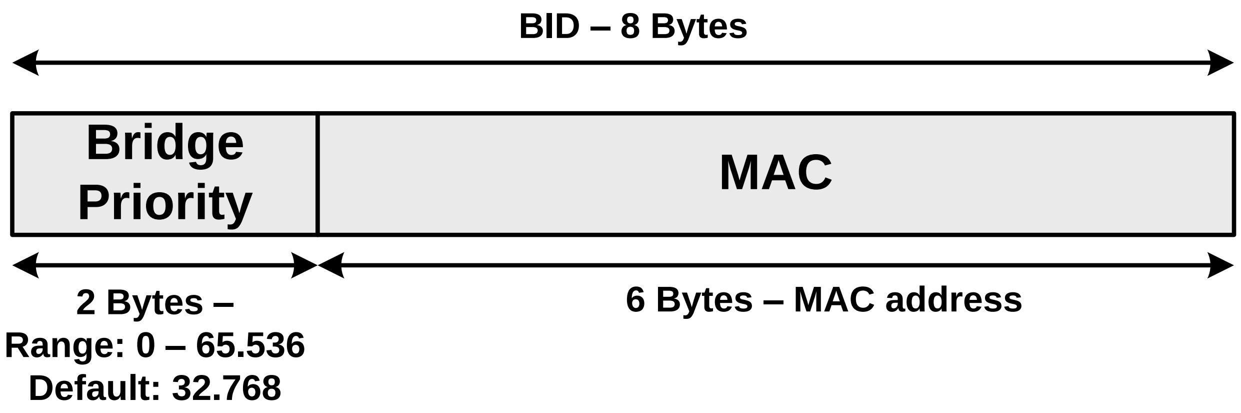
STP Variants ①

- **802.1D-1998**: The legacy standard for bridging and STP which is not any longer part of 802.1D standard
- **CST**: Assumes one spanning-tree instance for the entire bridged network, regardless of the number of VLANs
- **PVST+**: A Cisco enhancement of STP that provides a separate 802.1D spanning-tree instance for each VLAN configured in the 802.1Q network.
- **802.1D-2004**: An updated bridging and STP standard
- **802.1s** (**Multiple STP** or **MST** / **MSTP**): Maps multiple VLANs into the same spanning-tree instance
- **802.1w** (**Rapid STP** or **RSTP**): Improves convergence over 1998 STP by adding roles to ports and enhancing BPDU exchanges
- **PVRST+**: A Cisco enhancement of RSTP using PVST+
- **PVST**, **RPVST**: STP a RSTP instance for each VLAN in the ISL network

STP Variants ②

	Standard	Resources	Convergence	
CST	802.1D	Low	Slow	All VLANs
PVST+	Cisco	High	Slow	Per VLAN
RSTP	802.1w	Medium	Fast	All VLANs
PVRST+	Cisco	Very high	Fast	Per VLAN
MSTP	802.1s	Medium or high	Fast	VLAN list

STP Switch and Port Parameters



- Every switch has its own bridge ID (BID)
  - 2 B: Configurable priority
    - By default = 32768
  - 6 B: Switch’s MAC address

- Each port on switch has its own port ID (PID)
  - Divided into 2 parts just as BID
  - 1 B: Configurable priority
    - By default = 128
  - 1 B: Port Index

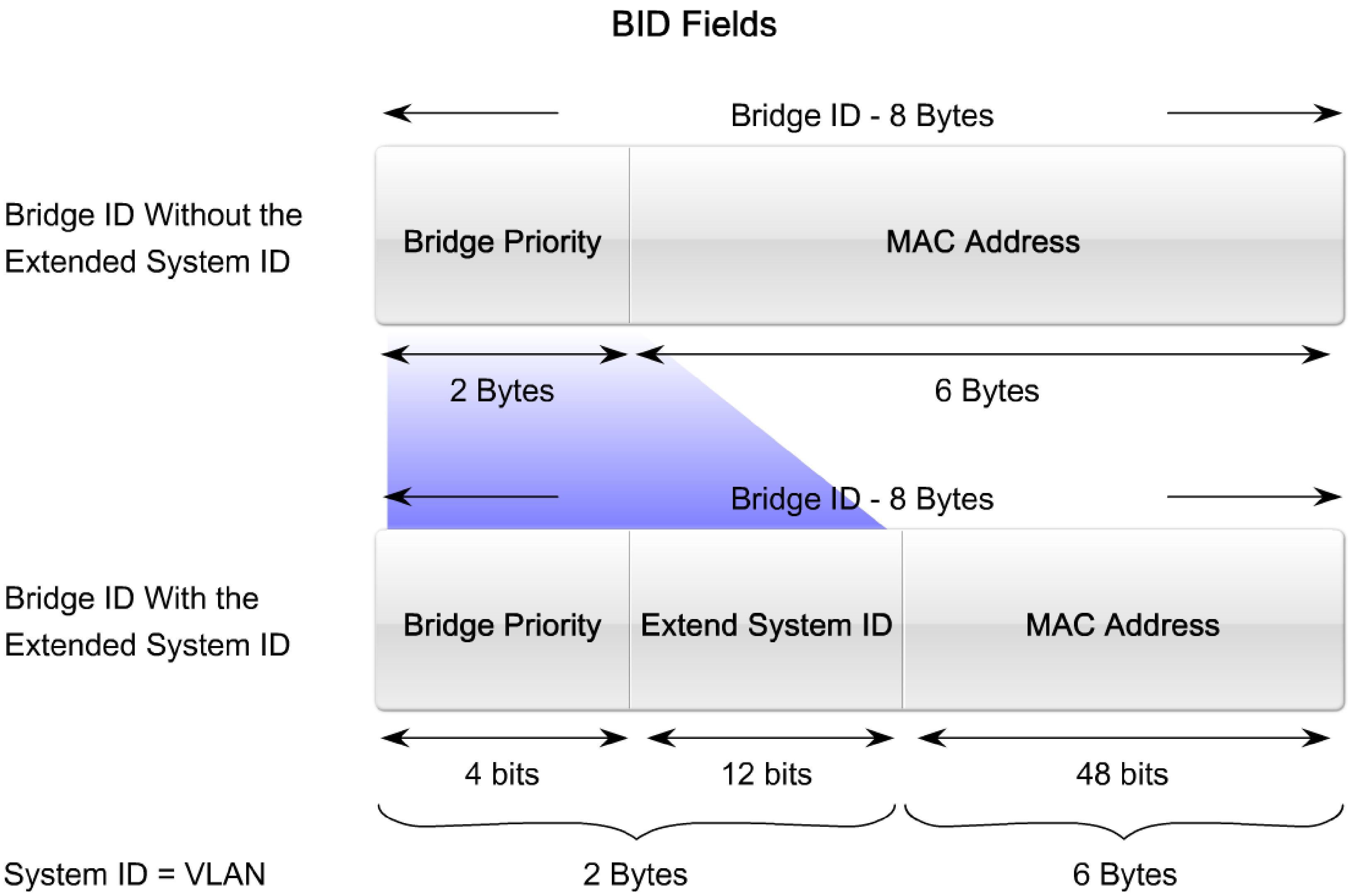
Extended Bridge ID ①

- 802.1D requires that BID of each switch in topology must be unique
  - To satisfy this each switch need some range of MAC address in networks as BID where are STP instances for each VLAN present
- Solution is Extended BID specified in IEEE 802.1t
  - Priority inside BID is split into two parts
    - Upper 4 bits: Configurable priority with steps of 4096
    - Next 12 bits: VLAN ID marking relevant VLAN to this STP instance
  - 802.1t is nowadays part of 802.1D-2004
- On older switches it was possible to turn usage of Extended BID on or off with following command:

```
Switch(config)# spanning-tree extend system-id
```

- Newer switches show this command in running configuration but it’s unable to turn it off

Extended Bridge ID ②



BID Fields

Bridge ID - 8 Bytes

Bridge ID Without the Extended System ID

Bridge Priority

MAC Address

2 Bytes

6 Bytes

Bridge ID - 8 Bytes

Bridge ID With the Extended System ID

Bridge Priority

Extend System ID

MAC Address

4 bits

12 bits

48 bits

System ID = VLAN

2 Bytes

6 Bytes

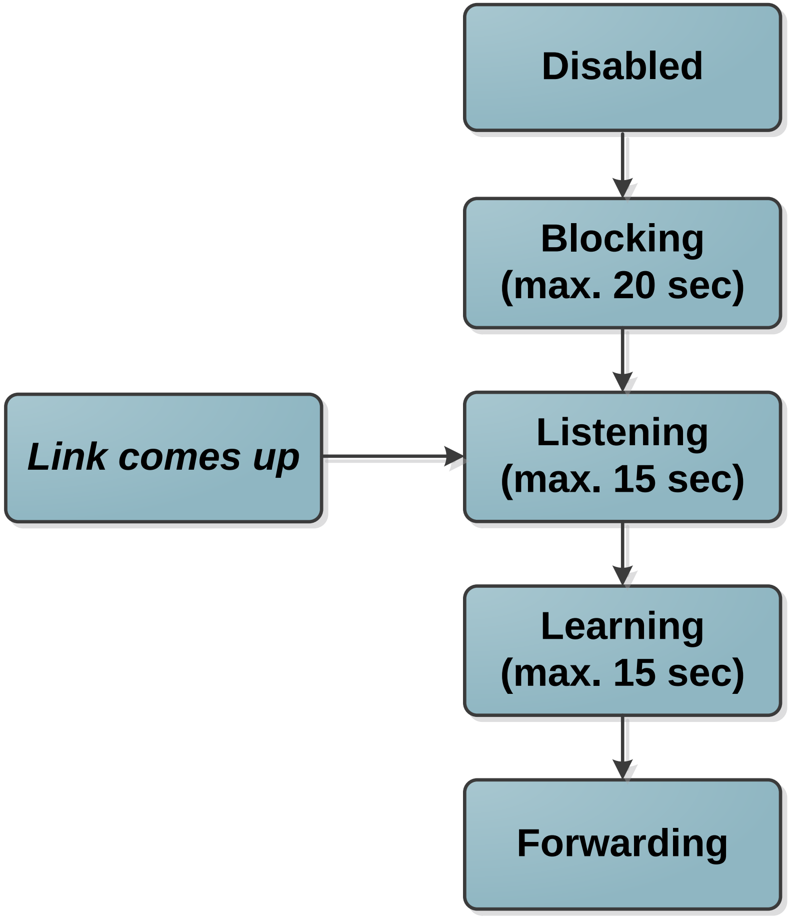
Port Cost

- Cost of link is tied up with every port/interface
  - It is derived from actual speed of interface by default
  - It is configurable
  - It is used for computation of distance between switch and root switch
- There are different types of default values according to speed:

Speed	802.1D-2004	802.1D-1998	Pre 802.1D-1998
10 Gbps	2000	2	1
1 Gbps	20 000	4	1
100 Mbps	200 000	19	10
10 Mbps	2 000 000	100	100

- Spanning Tree Protocol Default Port Cost

Legacy 802.1D STP: Port States



Disabled: Shutdown port or port which is not a part of any STP instance

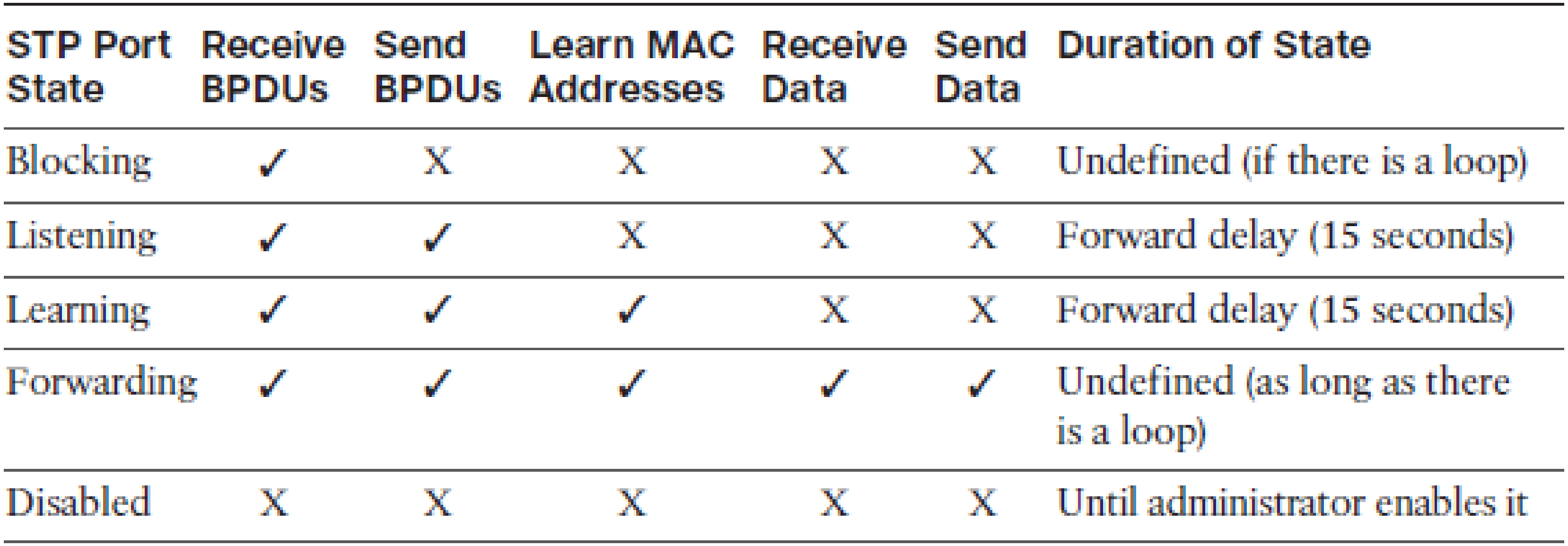
Blocking: Processes received BPDU but does not send own BPDUs. Neither send or receive data frames. Does not learn any MAC addresses.

Listening: Send and receive BPDU normally. Neither send or receive data frames. Does not learn any MAC addresses.

Learning: Send and receive BPDU normally. Neither send or receive data frames. Learn MAC addresses.

Forwarding: Send and receive BPDU normally. Send and receive data frames. Learn MAC addresses.

Legacy 802.1D STP: Port States



STP Port State	Receive BPDUs	Send BPDUs	Learn MAC Addresses	Receive Data	Send Data	Duration of State
Blocking	✓	X	X	X	X	Undefined (if there is a loop)
Listening	✓	✓	X	X	X	Forward delay (15 seconds)
Learning	✓	✓	✓	X	X	Forward delay (15 seconds)
Forwarding	✓	✓	✓	✓	✓	Undefined (as long as there is a loop)
Disabled	X	X	X	X	X	Until administrator enables it

Legacy 802.1D STP: Message Format

Bytes	Field
2	Protocol ID
1	Version
1	Message type
1	Flags
8	Root ID
4	Cost of path
8	Bridge ID
2	Port ID
2	Message age
2	Max age
2	Hello time
2	Forward delay

- Two BPDU Types
  - **Configuration BPDU**
    - Sent every 2 seconds
  - **Topology Change Notification (TCN) BPDU**

Who is root bridge?

How distant is root bridge?

What is BID of switch which generated this BPDU

Which port generated this BPDU?

- Initial RID which switch generate is its own BID. Root ID = Bridge ID

BPDU and STP Timers

- BPDUs are generated only by root bridge
  - Other switches just modify BPDU and resend it through other appropriate ports
  - Every switch remembers on each of its ports the best BPDU in last max_age seconds
- STP employs following timers:

Timer	Description
Hello Time	Time between two consecutive BPDUs. By default 2 seconds. Possible range: <1, 10>
Forward Delay	Amount of time which port remains in Listening and Learning state. By default 15 seconds. Possible range: <4, 30>
Maximum Age	Time during which port remembers the best received BPDU. By default 20 seconds. Possible range: <6, 40>

STP Port Roles

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

Decision Process when Comparing BPDUs

- STP builds its functionality on comparing BPDUs and ability to chose which one is better (superior) and which is worst (inferior)
- Decision compares subsequent parameters in following order:
  1. Root Bridge ID (has two parts)
  2. Root Path Cost
  3. Sender Bridge ID (has two parts)
  4. Sender Port ID (has two parts)
  5. Receiver Port ID (has two parts; only in rare cases its compared)
- Lower value is considered better
  - *Just as same as metric*

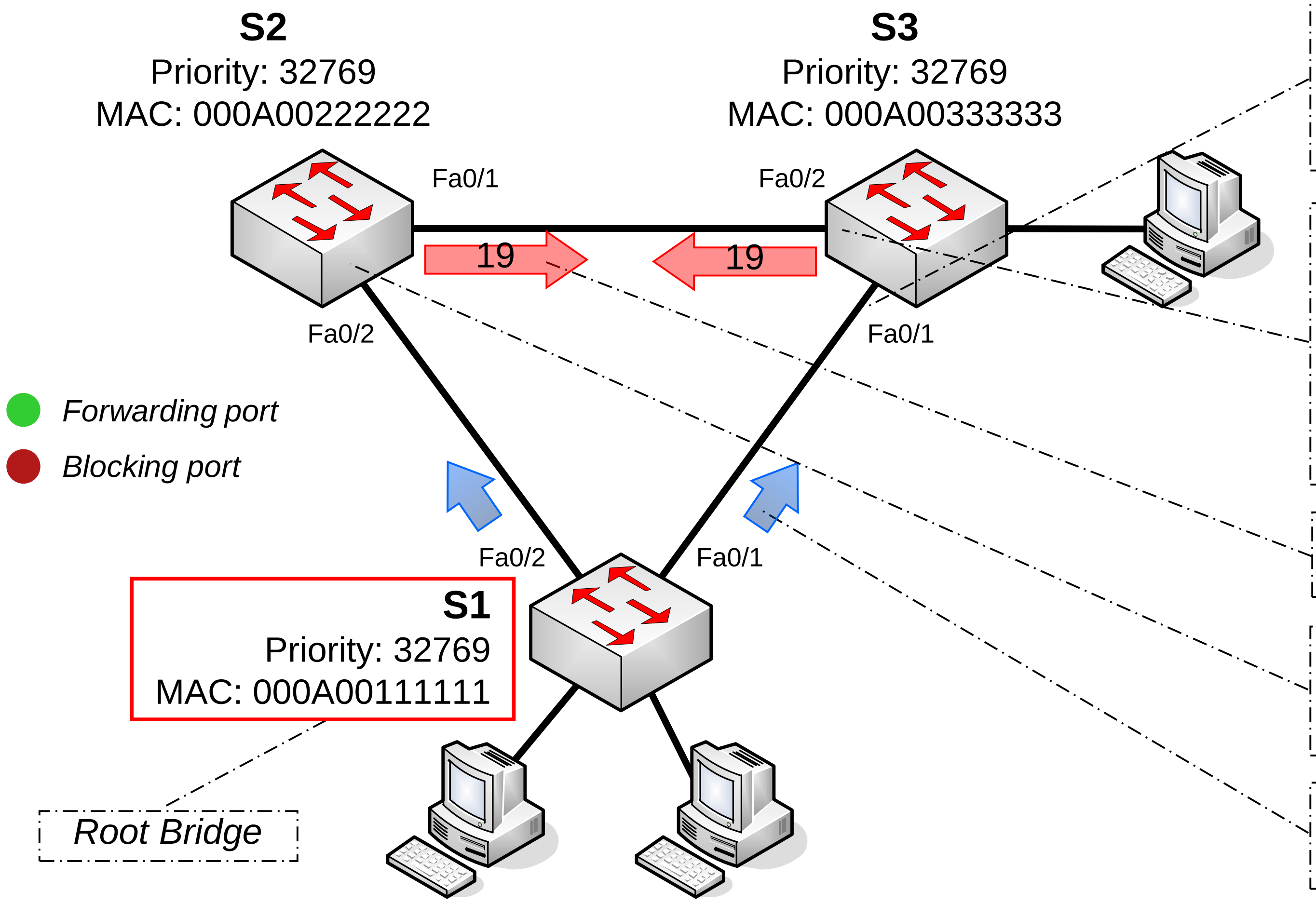
Spanning Tree Building Process ①

- Step 1: Election of root bridge
  - Root bridge is switch with the lowest BID
  - Initially every switch consider itself as a root before topology converge
  - `IF` switch receives superior BPDU `THEN` RBID is replaced with newer value
- Step 2: On each non-root bridge chose root port
  - Root port receives superior BPDU among all ports on target switch
  - Resulting BPDU (which is send by target switch): BPDU which root port cost was increased by a cost of port on which BPDU was received
    - e.g.: *BPDU received with RPC 19 on port with cost 19 has resulting BPDU with RPC cost 19 + 19 = 38*
  - Root port transfer through all state until it reaches Forwarding state

Spanning Tree Building Process ②

- Step 3: Election of Designated port on each segment
  - Designated port is port which own (send by switch) BPDU is superior among all other received BPDUs
  - BPDUs are compared immediately on reception before STP parameters alternation
  - Designated port transfer through all state until it reaches Forwarding state
- Except previous ones all other ports are useless and remains in Blocking state
- *What do we get after all?*
  - Some links between switches are unblocked on both endpoints
    - Designated port ↔ Root port
  - Other links between switches are unblocked only on the one endpoint
    - Designated port ↔ Non-designated port

Step 1 and Step 2 Examples



S2
Priority: 32769
MAC: 000A00222222

S3
Priority: 32769
MAC: 000A00333333

S1
Priority: 32769
MAC: 000A00111111

● *Forwarding port*

● *Blocking port*

Root Bridge

Among all received BPDUs root port is port with the most superior received BPDU.

Elected Root Ports

Switch S2 and S3 on ports:

Received RPC + Port cost
0+19=19

Resulting BPDUs on these ports are inferior; hence, they will not become root ports

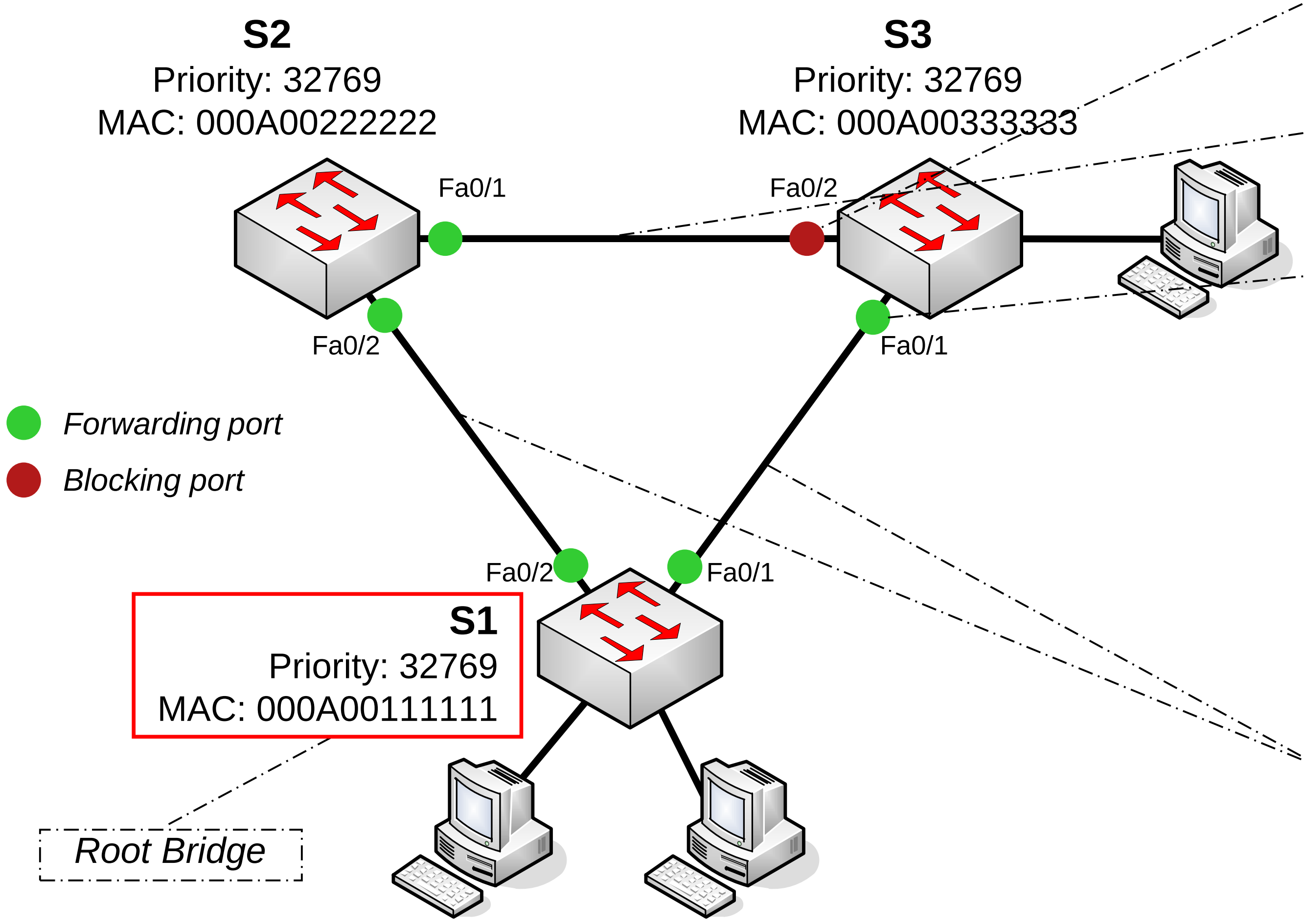
Switches passes BPDU with altered RPC value

Switch S2 and S3 on ports:

Received RPC + Port cost
0+19=19

Root Bridge sends BPDUs on each of its ports with RPC = 0.

Step 3 Examples



Designated port will be forwarding, other port will be blocking.

What roles will take following ports on link between S2 and S3?

Designated port is port on which all received BPDUs are inferior than the BPDU send through this port.

Decision process:

1. *Root Bridge ID*
2. *Root Path Cost*
3. *Sender Bridge ID*
4. *Sender Port ID*
5. *Receiver Port ID*

What roles will take following ports on links between S1-S2 and S1-S3?

The show spanning-tree Command on S1

```
S1# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     000A.0011.1111
              This bridge is the root
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769
              Address     000A.0011.1111
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- -------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Desg FWD 19        128.2    P2p
```

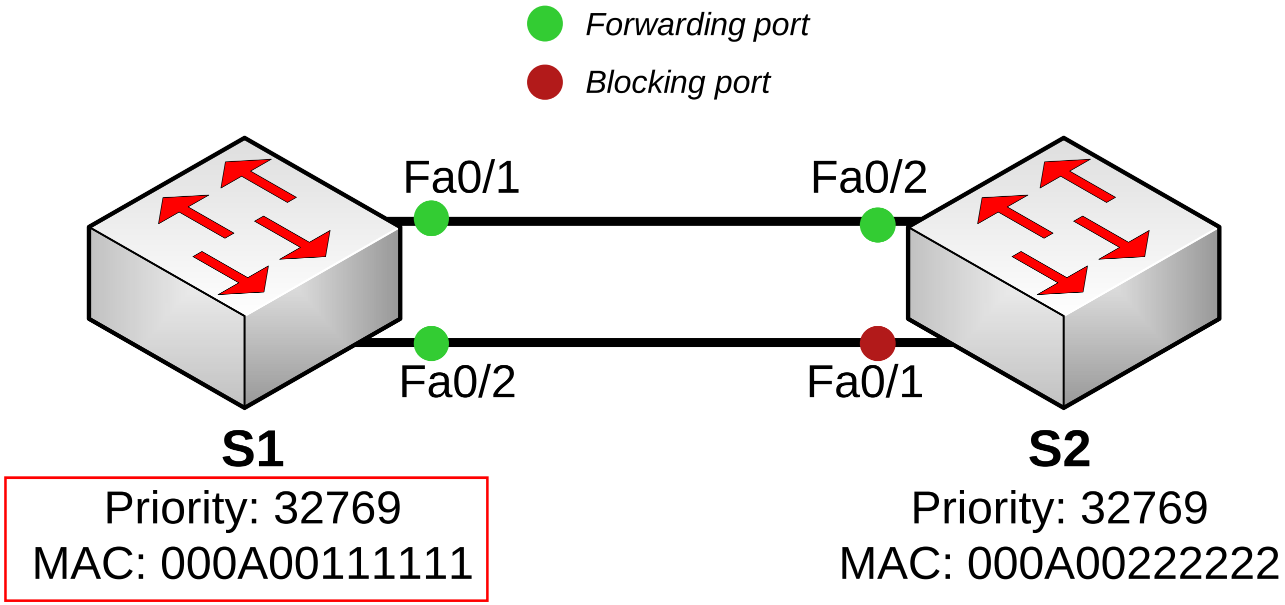
The show spanning-tree Command on S2

```
S2# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     000A.0011.1111
             Cost        19
             Port        2  (FastEthernet0/2)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769
             Address     000A.0022.2222
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- -------------------------
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Root FWD 19        128.2    P2p
```

The show spanning-tree Command on S3

```
S3# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     000A.0011.1111
             Cost        19
             Port        1 (FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769
             Address     000A.0033.3333
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- -------------------------
Fa0/2            Altn BLK 19        128.2    P2p
Fa0/1            Root FWD 19        128.1    P2p
```

STP Use-Case ①



Which switch is root bridge?

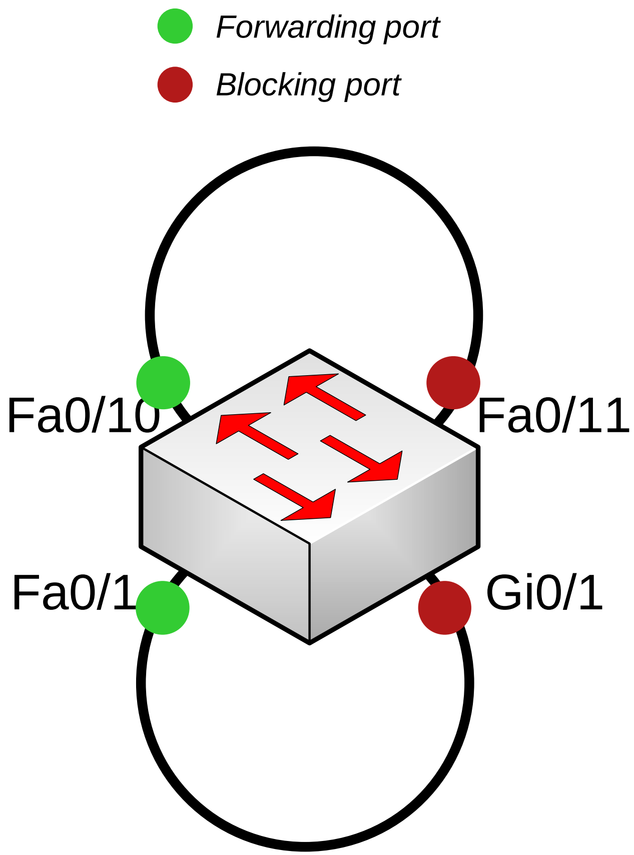
Which ports are designated?

Decision process:

1. *Root Bridge ID*
2. *Root Path Cost*
3. *Sender Bridge ID*
4. *Sender Port ID*
5. *Receiver Port ID*

Which ports are blocked and which are root ports?

STP Use-Case ②

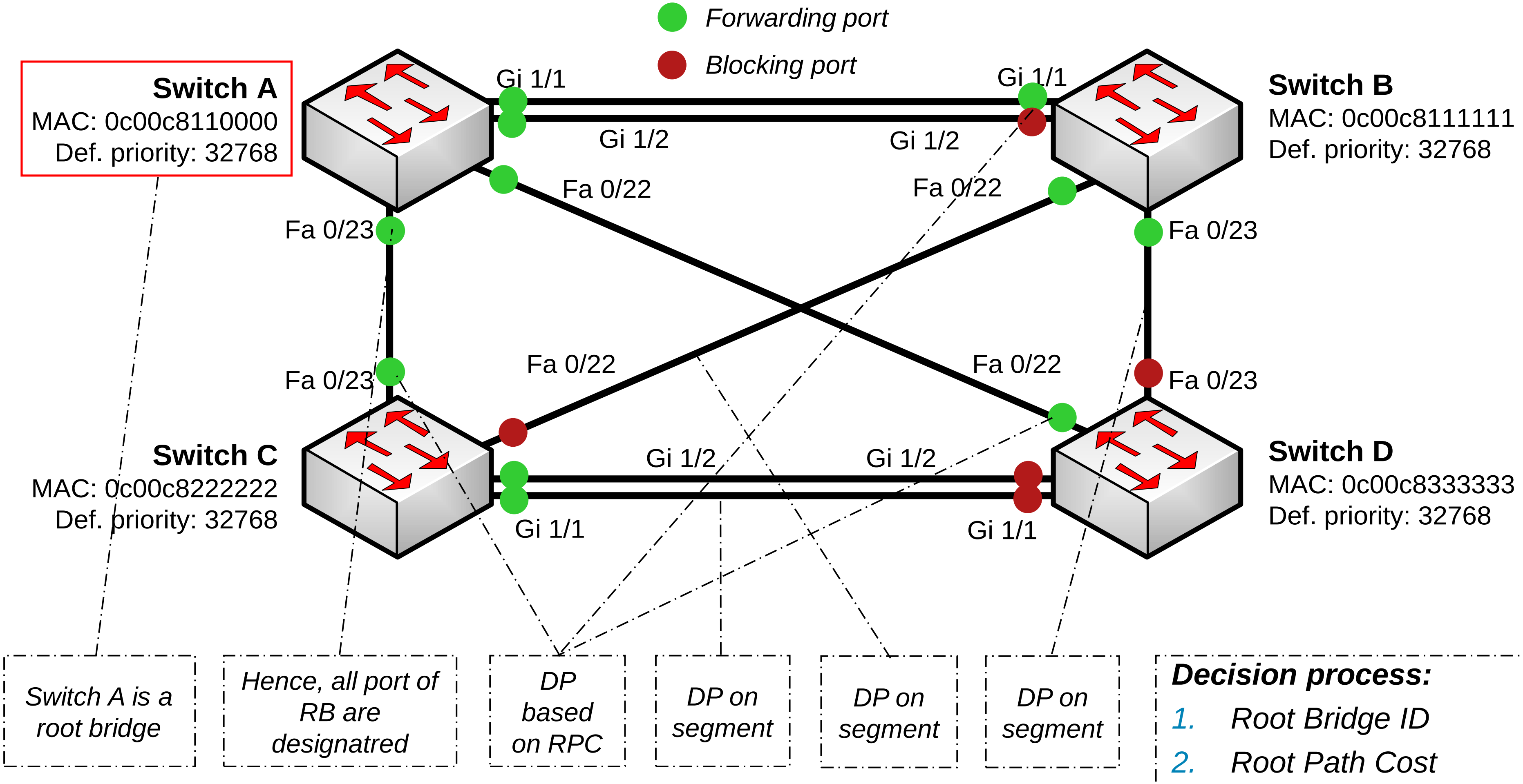


Decision process:

1. *Root Bridge ID*
2. *Root Path Cost*
3. *Sender Bridge ID*
4. *Sender Port ID*
5. *Receiver Port ID*

S1
Priority: 32769
MAC: 000A00111111

STP Use-Case ③



Switch A
MAC: 0c00c8110000
Def. priority: 32768

Switch B
MAC: 0c00c8111111
Def. priority: 32768

Switch C
MAC: 0c00c8222222
Def. priority: 32768

Switch D
MAC: 0c00c8333333
Def. priority: 32768

Switch A is a root bridge

Hence, all port of RB are designatred

DP based on RPC

DP on segment

DP on segment

DP on segment

Decision process:

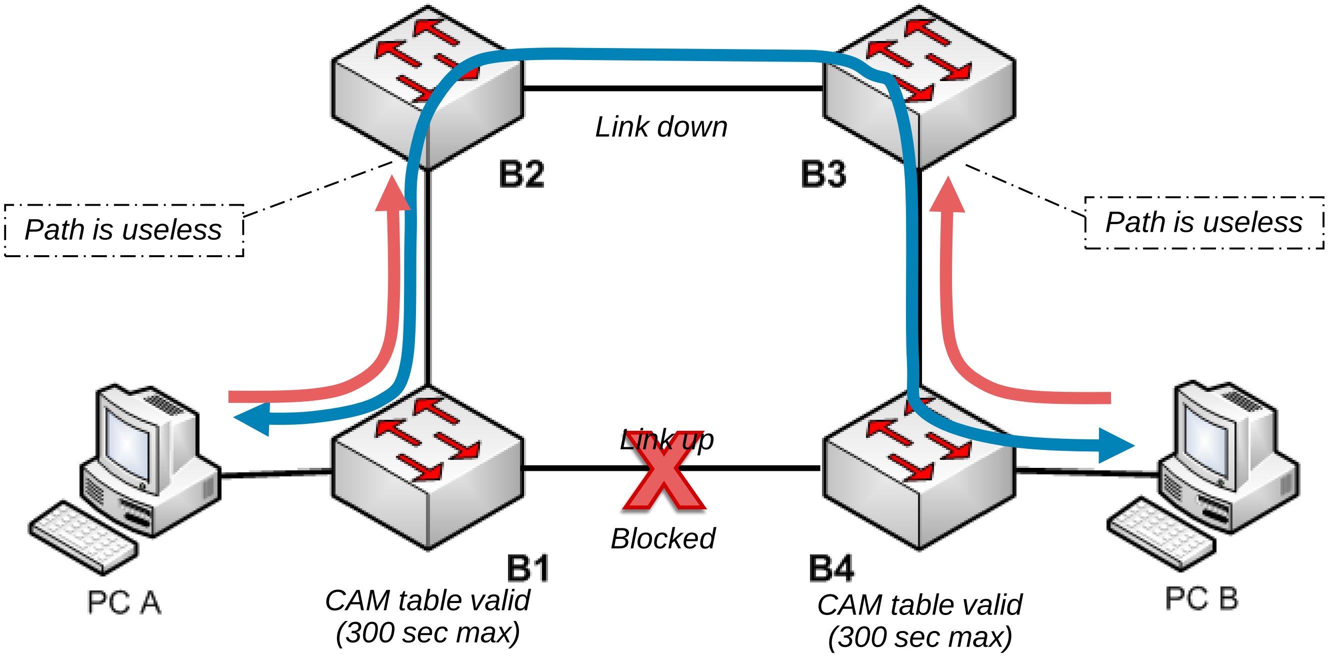
1. *Root Bridge ID*
2. *Root Path Cost*
3. *Sender Bridge ID*
4. *Sender Port ID*
5. *Receiver Port ID*

Converged STP Network

- Network where STP have found spanning tree has following properties
  - L2 loops are eliminated
  - 1 Root Bridge per VLAN
  - 1 root port on every non-root switch
  - 1 designated port on every segment
  - Non-designated ports are blocking
- STP algorithm never finishes – it is recalculated upon every receive of BPDU

Topology Change in STP

- In stable topology path between PC A and PC B leads via switches B1, B2, B3, B4
  - CAM table is also stable, records expire after 300 seconds after no refresh



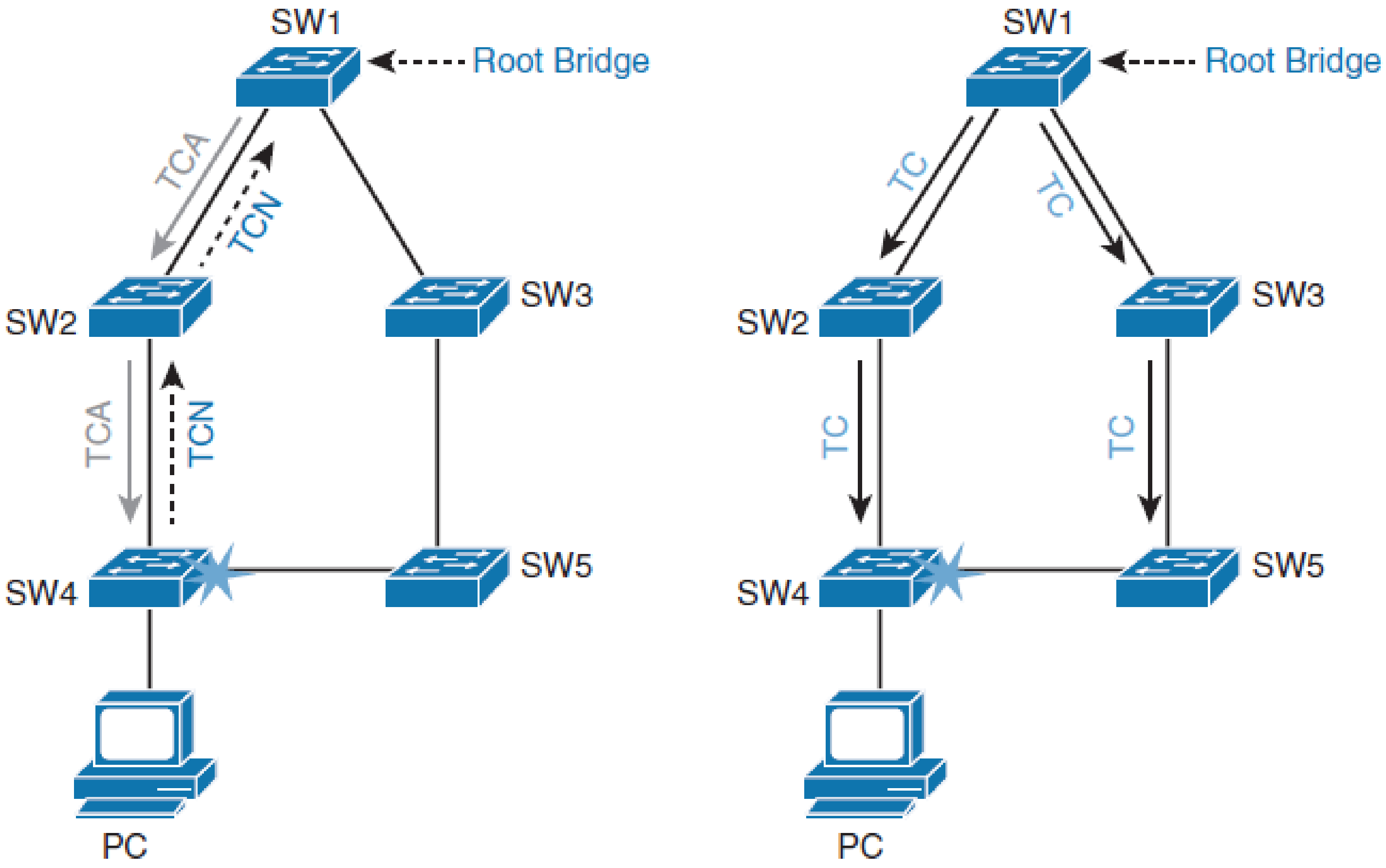
- *What could be solution?*
  - Quicker CAM records expiration

Topology Change Notification ①

Topology change = switchport changes state to Forwarding or Blocking.

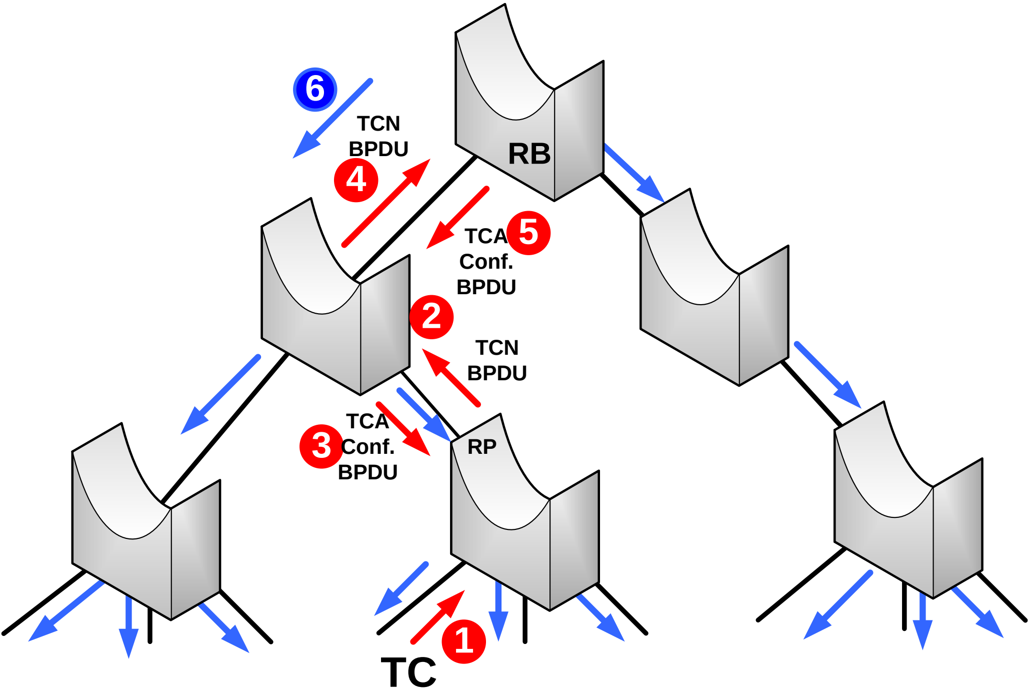
Switch announce topology change to the root bridge.

Root bridge signals the topology change to other switches.



Topology Change Notification ②

1) Switch sends TCN BPDU in two cases `IF`:
   - port transits to forwarding state
   - port transits from state Forwarding to state Learning or Blocking
2) Uplink switch receives TCN BPDU through its designated port
3) Uplink switch sets Topology Change Acknowledgment Flag (TCA) in next configuration BPDU and sends it back to downlink switch
   - Downlink switch stops generating TCN BPDU upon receiving TCA
4) Uplink switch sends TCN BPDU via root port towards Root Bridge



5) Steps 2), 3) and 4) are repeated until root bridge receives TCN BPDU after that Root Bridge confirms reception with TCA
6) Root Bridge sets TC Flag in the next configuration BPDU and periodically sends it for the next Forward Delay + Max Age seconds (by default 35 seconds)
7) All switches in topology receive configuration BPDU with set TC flag
   - TC flag tells switches to shorten expiration of CAM table records to the Forward Delay value (by default 15 s)

- Document ID 12013: “Understanding Spanning-Tree Protocol Topology Changes”

Configuring 802.1D-2004 STP

Instances and Priority

```
! Following command to enable STP is not necessary
Switch(config)# spanning-tree vlan vlan-id

! Disable STP for target VLAN
Switch(config)# no spanning-tree vlan vlan-id
```

```
! Setting STP priority for target VLAN
Switch(config)# spanning-tree vlan vlan-id priority PRIORITY

% Allowed values are:
   0 4096 8192 12288 16384 20480 24576 28672
   32768 36864 40960 45056 49152 53248 57344 61440
```

Root Priority Macros

```
! Macro for setting root bridge
! IF current root bridge has priority > 24576
   THEN it sets local priority to 24576
! IF current root bridge has priority < 24576
   THEN it sets local priority to priority - 4096


Switch(config)# spanning-tree vlan vlan-id root primary
```

```
! Macro for setting backup root bridge
! Sets priority to the fixed value 28672


Switch(config)# spanning-tree vlan vlan-id root secondary
```

Problem when Using Macro

```
DLS1(config)# do show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority     4097
             Address      0017.9446.ad00
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
DLS2(config)# spanning-tree vlan 1 root primary
% Failed to make the bridge root for vlan 1
% It may be possible to make the bridge root by setting the priority
% for some (or all) of these instances to zero.
```

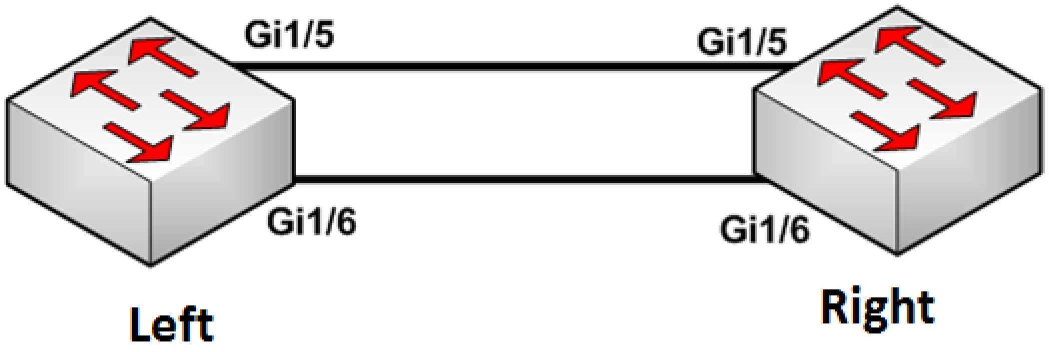
STP Tuning: Cost and Port Priority

```
Switch(config)# interface range fa 0/11 - 12
! Changing port cost for all VLANs
Switch(config-if-range)# spanning-tree cost 32
! Changing port cost for target VLAN
Switch(config-if-range)# spanning-tree vlan vlan-id cost 32
```

```
! Changing port priority, by default 128, range 0 - 240
Switch(config)# interface range fa 0/11 - 12
! Changing port priority for all VLANs
Switch(config-if-range)# spanning-tree port-priority 112
! Changing port priority for target VLAN
Switch(config-if-range)# spanning-tree port-priority 112
```

```
DLS2(config-if)# span vlan 1 port-priority ?
  <0-240>   port priority in increments of 16

DLS2(config-if)# span vlan 1 port-priority 112
DLS2(config-if)# do sh span vlan 1 int fa 0/7

Vlan                 Role Sts Cost      Prio.Nbr Type
-------------------- ---- --- --------- -------- --------------------------------
VLAN0001             Desg FWD 19        112.9    P2p
```

Port Priority Use-Case



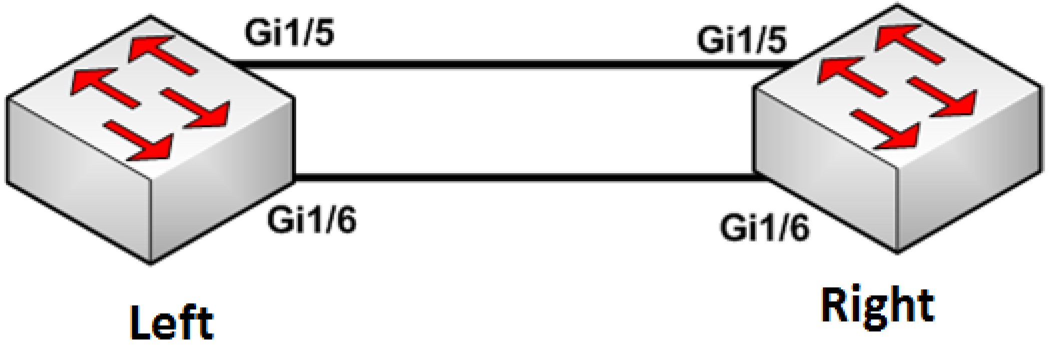
- Left is root bridge
  - *We have access to it*
- Right has root port on Gi1/5
  - *We do not have access to it*
- *How can we ensure that root port will be Gi1/6?*

```
Left(config)# interface gi1/6
Left(config-if)# spanning-tree port-priority 112
```

And what if we need to change it for VLAN 30 and trunk?

```
Left(config)# interface gi1/6
Left(config-if)# spanning-tree vlan 30 port-priority 112
```

Cost Use-Case



- Left is root bridge
  - *We have access to it*
- Right has root port on Gi1/5
  - *We do not have access to it*
- *How can we ensure that port will be Gi1/6 not blocking?*

```
Left(config)# interface gi1/6
Left(config-if)# spanning-tree cost 1
```

And what if we need to change it for VLAN 30 and trunk?

```
Left(config)# interface gi1/6
Left(config-if)# spanning-tree vlan 30 cost 1
```

Notes on Priority and Cost Tuning

- Tasks considering STP manipulation with priority or cost could be solved in two ways:
    1. By decrementing value of priority or cost on preferred object
    2. By incrementing value of priority or cost on NON-preferred object
- It is usually more appropriate to increment cost on target interfaces – use second way
    - Because usually on target interface it could lead to deadlock where is no chance for improvement (there’s no way how to decrement bellow value 1)
- *It is matter of praxis to chose what is suitable more and what less for current situation!*

Tuning STP: Timers

- Document ID 19120: „Understanding and Tuning Spanning Tree Protocol Timers“

```
! STP timers configuration
Switch(config)#
   spanning-tree [vlan vlan-id] hello-time seconds
   spanning-tree [vlan vlan-id] forward-time seconds
   spanning-tree [vlan vlan-id] max-age seconds
```

```
! Automatic correction of timers according to diameter
Switch(config)#
  spanning-tree vlan vlan-list root { primary | secondary }
  [diameter DIAMETER [hello-time HELLO-TIME]]
```

The show spanning-tree Commands ①

```
Switch# show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     0c00c8110000
             Cost        19
             Port        1  (FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
             Address     0c00c8111111
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300
<text omitted>
```

```
! Display information only to target VLAN
Switch# show spanning-tree vlan vlan-id
```

```
Switch# show spanning-tree interface FastEthernet 0/7

Vlan                Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----------------------------
VLAN0001            Altn BLK 19        128.7    P2p
VLAN0100            Root FWD 19        128.7    P2p
VLAN0110            Root FWD 19        128.7    P2p
VLAN0120            Root FWD 19        128.7    P2p
```

The show spanning-tree Commands ②

```
! Show all detail informations
Switch# show spanning-tree detail

VLAN0001 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 4096, sysid 1, address 0017.9446.ad00
  Configured hello time 2, max age 20, forward delay 15
  Current root has priority 1, address 0017.9460.3080
  Root port is 13 (FastEthernet0/11), cost of root path is 19
  Topology change flag not set, detected flag not set
  Number of topology changes 15 last change occurred 00:35:32 ago
          from FastEthernet0/1
  Times:  hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0, aging 300

 Port 9 (FastEthernet0/7) of VLAN0001 is designated forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.9.
   Designated root has priority 1, address 0017.9460.3080
   Designated bridge has priority 4097, address 0017.9446.ad00
   Designated port id is 128.9, designated path cost 19
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   BPDU: sent 18544, received 25
```

```
! Previous command limited only on target interface
Switch# show spanning-tree interface fastethernet 0/7 detail
```

The show spanning-tree Commands ③

```
! Display Root Bridge ID, Root Port and Root Path Cost
Switch# show spanning-tree vlan vlan-id root
```

```
DLS1# show spanning-tree vlan 1 root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Root Port
---------------- -------------------- --------- ----- --- ---  ------------
VLAN0001             1 0017.9460.3080        19    2   20  15  Fa0/11
```

```
! Display Bridge ID and timers
Switch# show spanning-tree vlan vlan-id bridge
```

```
DLS1# show spanning-tree vlan 1 bridge

                                                   Hello  Max  Fwd
Vlan                         Bridge ID              Time  Age  Dly  Protocol
---------------- --------------------------------- -----  ---  ---  --------

VLAN0001          4097 ( 4096,   1) 0017.9446.ad00    2    20   15  ieee
```

The debug Command ①

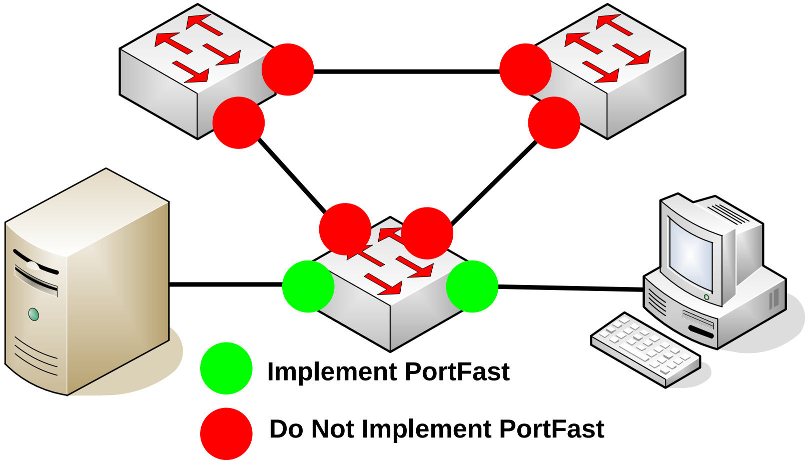
```
DLS1# debug spanning-tree ?
  all                  All Spanning Tree debugging messages
  backbonefast         BackboneFast events
  bpdu                 Spanning tree BPDU
  bpdu-opt             Optimized BPDU handling
  config               Spanning tree config changes
  csuf/csrt            STP CSUF/CSRT
  etherchannel         EtherChannel support
  events               Spanning tree topology events
  exceptions           Spanning tree exceptions
  general              Spanning tree general
  mstp                 MSTP debug commands
  pvst+                PVST+ events
  root                 Spanning tree root events
  snmp                 Spanning Tree SNMP handling
  switch               Switch Shim debug commands
  synchronization      STP state sync events
  uplinkfast           UplinkFast events
```

The debug Command ②

```
Switch# debug spanning-tree events
Spanning Tree event debugging is on
22:32:23: set portid: VLAN0001 Fa0/6: new port id 800D
22:32:23: STP: VLAN0001 Fa0/6 -> listening
22:32:25: %LINK-3-UPDOWN: Interface FastEthernet0/6, changed state to up
22:32:26: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/6,
changed state to up
22:32:38: STP: VLAN0001 Fa0/6 -> learning
22:32:53: STP: VLAN0001 Fa0/6 -> forwarding
```

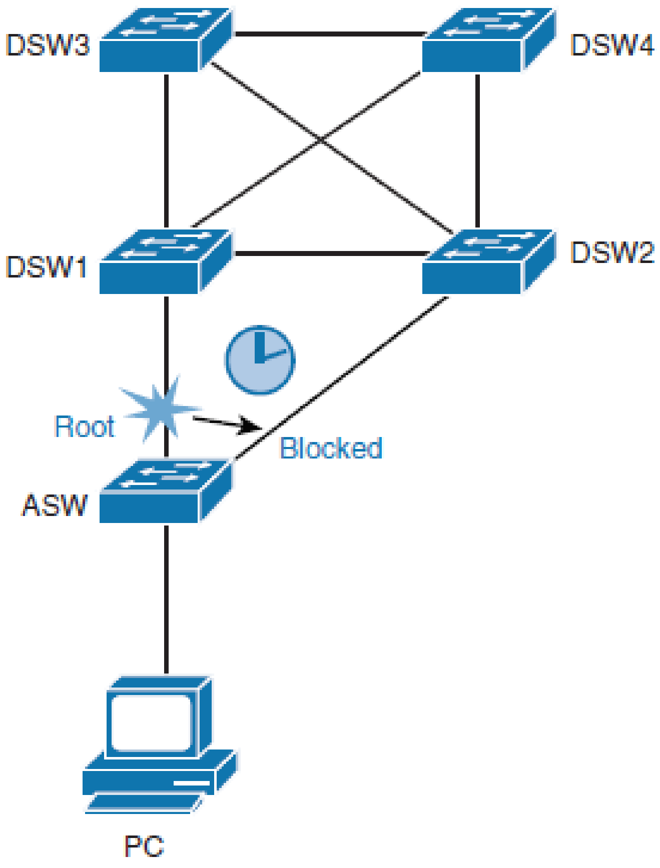
Cisco Enhancement to STP

- **PortFast**
  - Port transits immediately to Forwarding state
  - Usable only for access port
  - `IF` PortFast port receives BPDU `THEN` PortFast is disabled on that port and begins to behave as standard STP port
  - Enabling/Disabling PortFast does not cause generating of TCN



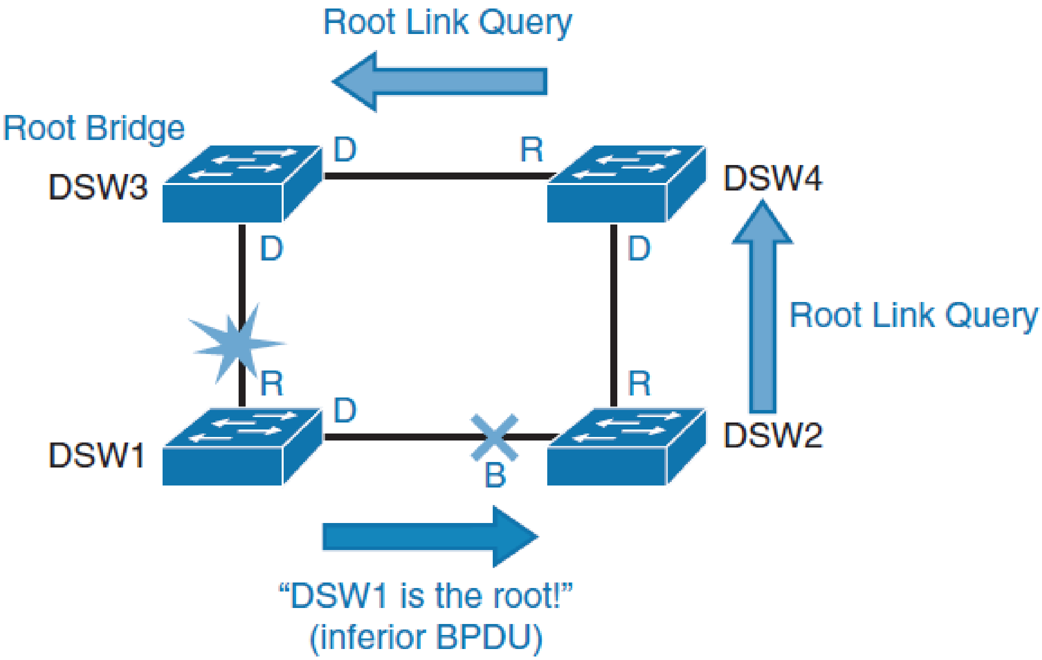
- **UplinkFast** ([Document ID 10575])
  - Speeds up activation of another Root Port in the case of failure of the current one
  - Only on non-root bridges
- **BackboneFast** ([Document ID 12104])
  - Speeds up convergence whenever other switch’s Root Port went down
- **PVST/PVST+**
  - Separate STP instance for each VLAN
  - PVST BPDUs are encapsulated same as STP BPDUs
- RSTP takes best of UplinkFast and BackboneFast features and adopts it

Use UplinkFast



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

Use BackboneFast



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

Use BackboneFast

- Normally a switch must wait for the maximum age timer to expire before responding to the inferior BPDUs.
- However BackboneFast searches for an alternative path:
  - `IF` the inferior BPDU arrives on a port that is blocked `THEN` the switch assumes that the root port and all other blocked ports are an alternative path.
  - `IF` the inferior BPDU arrives on a port that is root `THEN` the switch assumes all blocked are an alternate path.
  - `IF` no ports are blocked `THEN` the switch assumes that it lost connectivity with the root bridge `AND` considers itself as the root bridge.
- After the switch identifies potential alternative ports, it starts sending RLQs (request link queries). By sending these queries, it finds out whether upstream switches have a path to the root bridge.
- To configure BackboneFast, use the following command:
  - `DSW1(config)#` **`spanning-tree backbonefast`**
- By default, BackboneFast is disabled.
- To verify the current BackboneFast state, issue the following command:
  - `DSW1#` **`show spanning-tree backbonefast`**
  - `BackboneFast is enabled`
- BackboneFast was implemented into RSTP
  - RSTP implementation differs a bit from BackboneFast
  - Whereas BackboneFast relies on RLQ messages to validate the current root bridge, RSTP relies on cached information.

Cisco STP: PortFast

```
! Enable PortFast on all access ports
Switch(config)# spanning-tree portfast default
```

```
! PortFast configuration per interface (doesn’t apply on trunks)
Switch(config)# int range fa 0/1 - 10
Switch(config-if)# spanning-tree portfast
```

```
! Disable PortFast on interfaces whenever it is enabled globally
Switch(config)# int range fa 0/1 - 10
Switch(config-if)# spanning-tree portfast disable
```

```
! Enabling PortFast on trunk port
! Used only towards routers and servers, never towards switches!!!
Switch(config)# int fa0/24
Switch(config-if)# spanning-tree portfast trunk
```

```
! Veryfing PortFast capability on interface
ALS1# show spanning-tree interface fa 0/1 portfast
VLAN0100            enabled
```

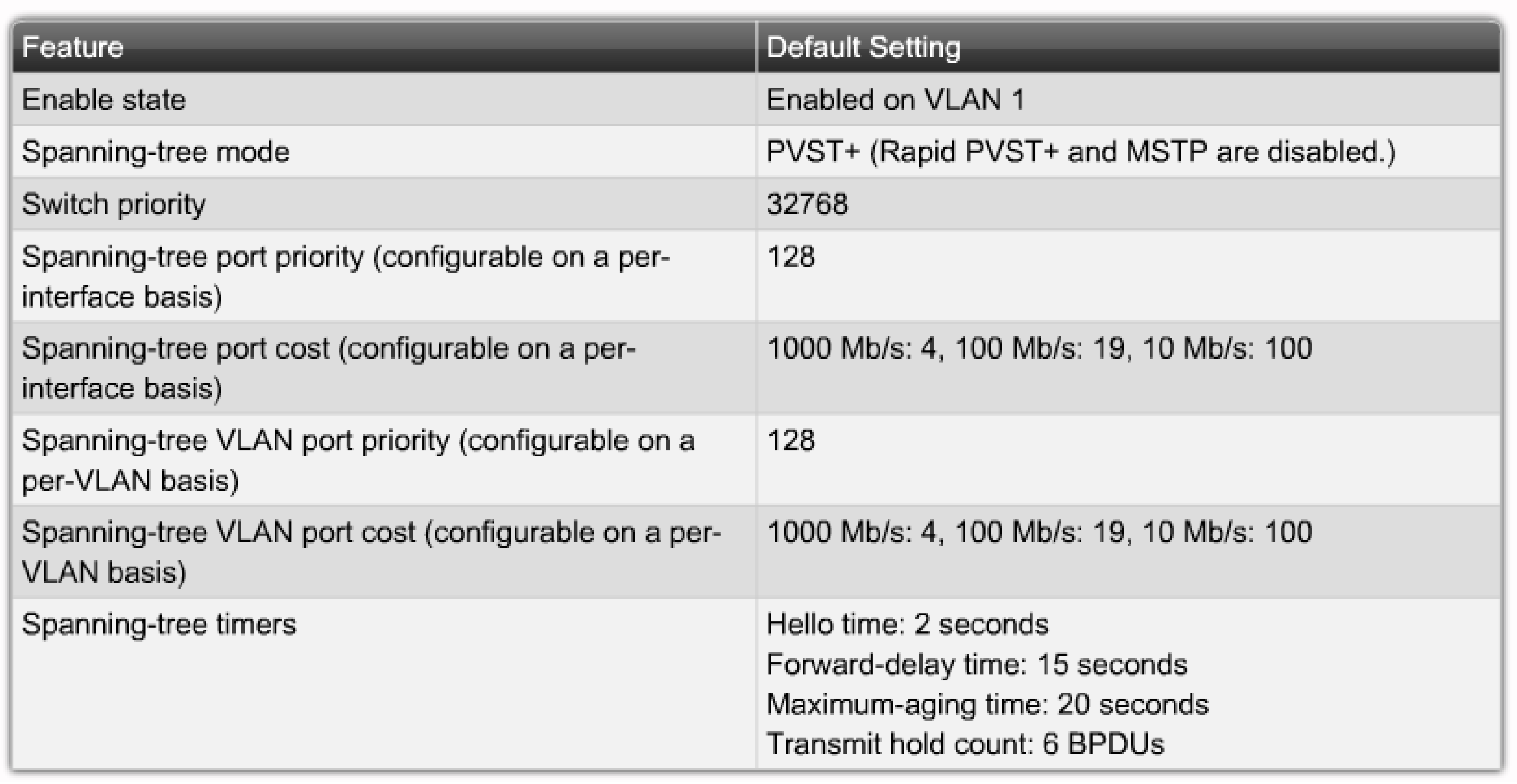
Verifying PortFast

```
Switch# show spanning-tree summary

Root bridge for:VLAN0001
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast             is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard            is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long
```

Host Macro

```
ALS1(config)# int fa 0/2
ALS1(config-if)# switchport host

switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled
```

Default Catalyst Settings



Feature	Default Setting
Enable state	Enabled on VLAN 1
Spanning-tree mode	PVST+ (Rapid PVST+ and MSTP are disabled.)
Switch priority	32768
Spanning-tree port priority (configurable on a per-interface basis)	128
Spanning-tree port cost (configurable on a per-interface basis)	1000 Mb/s: 4, 100 Mb/s: 19, 10 Mb/s: 100
Spanning-tree VLAN port priority (configurable on a per-VLAN basis)	128
Spanning-tree VLAN port cost (configurable on a per-VLAN basis)	1000 Mb/s: 4, 100 Mb/s: 19, 10 Mb/s: 100
Spanning-tree timers	Hello time: 2 seconds Forward-delay time: 15 seconds Maximum-aging time: 20 seconds Transmit hold count: 6 BPDUs

Rapid STP

Rapid Spanning Tree (RSTP)

- Document ID 24062: “Understanding Rapid Spanning Tree Protocol (802.1w)”
- RSTP is the next evolutional step of STP which improves stability, reliability and significantly increases speed
  - Reaction time bellow 1 second
  - Backward compatible with Legacy STP (on per-port basis)
  - Preferable choice for full-duplex switching environment
- Differences from legacy STP
  - New definition of port states and port roles
  - Link types and port functions
  - Proposal/Agreement Mechanism
  - Open versions of UplinkFast and BackboneFast
  - BPDU are generated by every switch

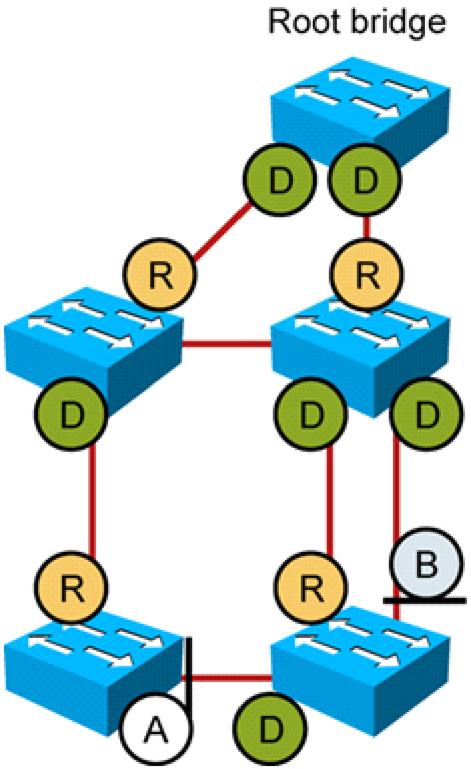
RSTP Port States

- **Discarding** (replacement for Blocking/Listening in STP)
  - Doesn’t forward data frames
- **Learning** (same as in STP)
  - Populating CAM table with MAC addresses
- **Forwarding** (same as in STP)
  - Fully operational port

Operational Status	STP Port State	RSTP Port State	Port in Topology	Port Learning MAC Addresses
Enable	Blocking	**Discarding**	No	No
Enable	Listening	**Discarding**	No	No
Enable	Learning	**Learning**	Yes	Yes
Enable	Forwarding	**Forwarding**	Yes	Yes
Disable	Disable	**Discarding**	No	No

RSTP Port Roles ①

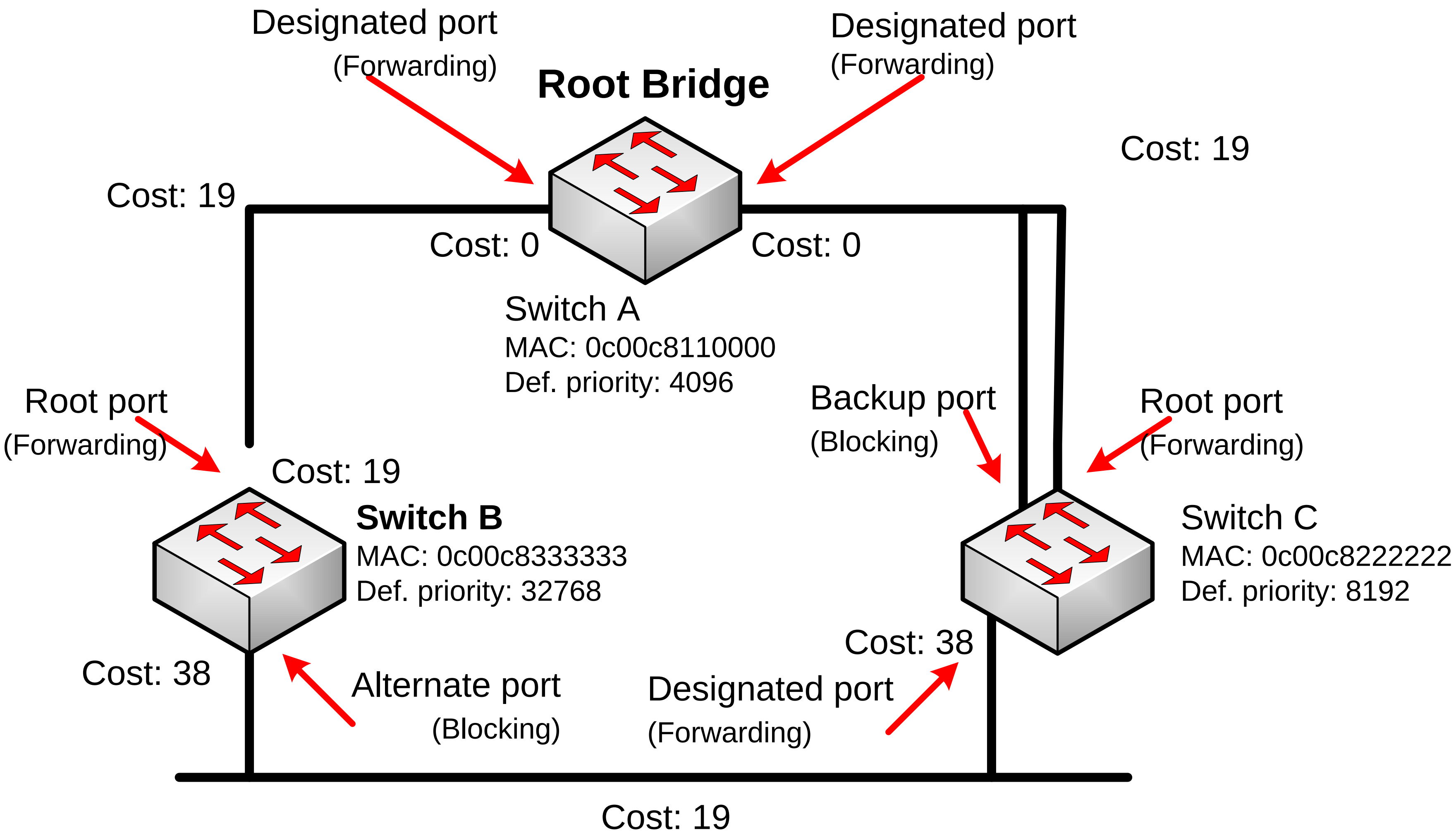
Port Role	Description
Root port	Port with best path to root bridge. It is in Forwarding state in stable topology.
Designated port	Port sending the best BPDU on the segment to which it is connected. It is in Forwarding state in stable topology.
Alternate port	Receives more useful BPDUs from another bridge. In other words standby port for current Root port. It is in Discarding state in stable topology.
Backup port	Receives more useful BPDUs from the same bridge. It is in Discarding state in stable topology.



RSTP Port Roles ②

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	Alternative or Backup port	Blocking	Discarding
Disabled	Disabled	—	Discarding
Transition	Transition	Listening Learning	Learning

Port Roles Use-Case



Root Bridge

Designated port
(Forwarding)

Designated port
(Forwarding)

Cost: 19

Cost: 19

Cost: 0

Cost: 0

Switch A
MAC: 0c00c8110000
Def. priority: 4096

Root port
(Forwarding)

Cost: 19

Switch B
MAC: 0c00c8333333
Def. priority: 32768

Backup port
(Blocking)

Root port
(Forwarding)

Switch C
MAC: 0c00c8222222
Def. priority: 8192

Cost: 38

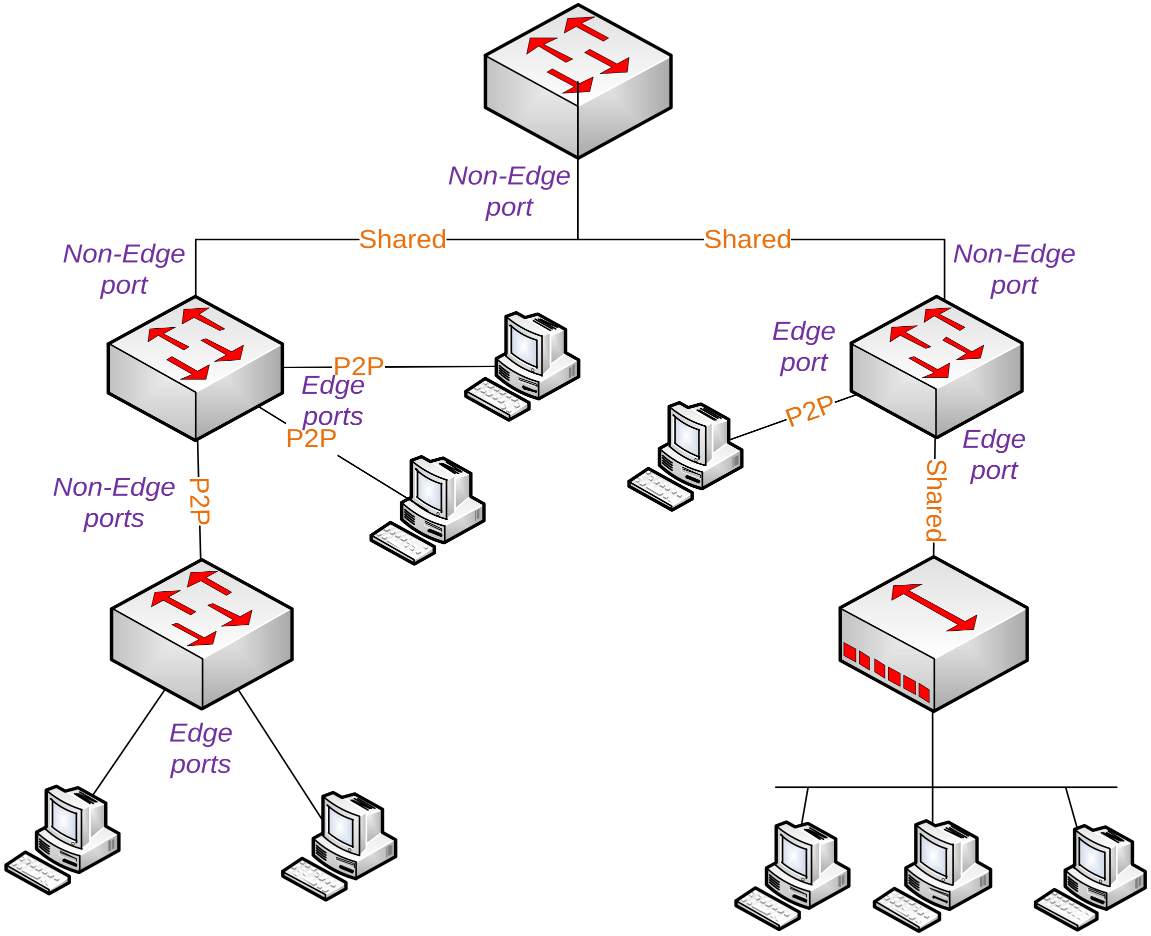
Alternate port
(Blocking)

Designated port
(Forwarding)

Cost: 38

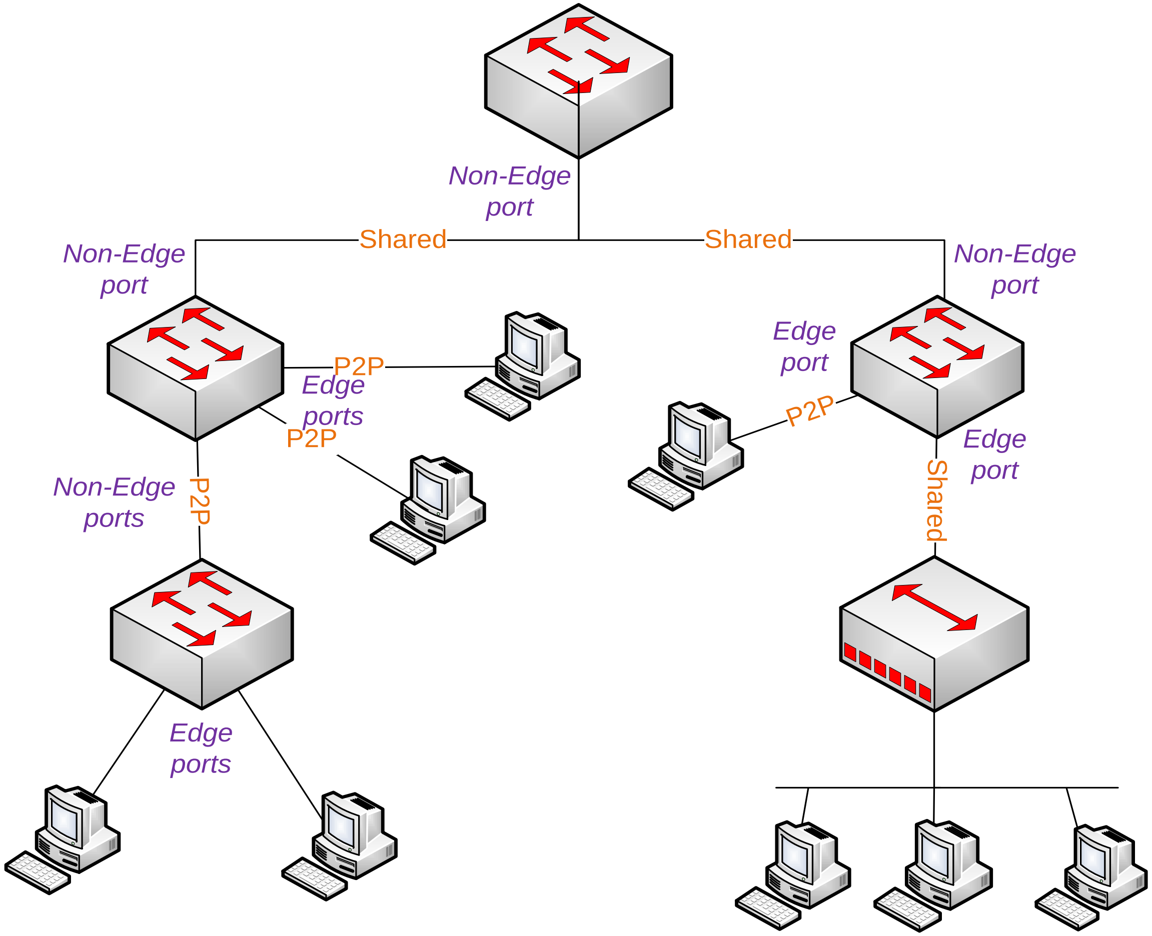
Cost: 19

RSTP Port Types



- **Edge Port**
  - Leads towards end host (PC, server) and will never have a switch connected to it
  - Immediately transits to Forwarding state
  - Does not cause topological changes
  - `IF` port receives BPDU `THEN` it becomes standard RSTP port
  - Is configured same as PortFast (no automatic detection)
- **Non-Edge port**
  - Internal port between switches
  - Default port type

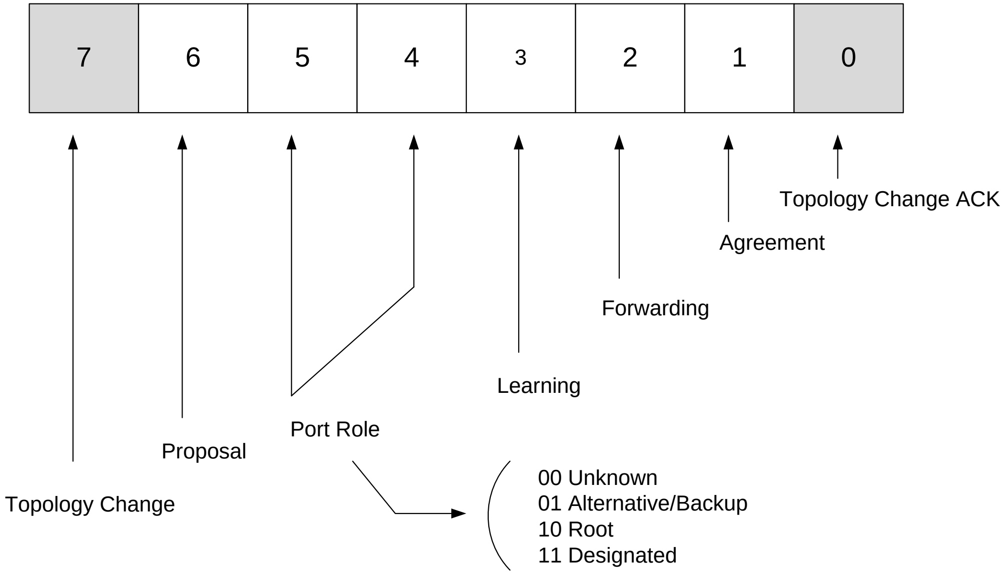
RSTP Link Types



- **Point-to-Point**
  - Port is connected to a single switch or other device at the other end of the link
  - Port operating in full-duplex mode
  - Uses Proposal/Agreement Mechanism
- **Shared**
  - Port operating in half-duplex mode
  - It is assumed that the port is connected to shared media where multiple switches might exist
  - Must transits through all states using proper timers settings
- Link type could be configured manually

BPDU v2: Flag Byte

- BPDU are sent every 2 seconds
  - Every switch generate own appropriate BPDUs, not just relaying the one generated by root bridge
  - BPDU serves same purpose as Hello mechanism
  - 3 missing BPDUs means topology change
- Bits:
  - 0 and 7 are reserved for TCN and TCA from 802.1D
  - 1 and 6 are used for Proposal/Agreement mechanism
  - 2 – 5 show port role and port state of the port which generated this BPDU



RSTP Topology Changes

- With RSTP, the TC propagation is now a one-step process
  - In fact, the initiator of the topology change floods this information throughout the network, as opposed to 802.1D, where only the root did
  - This mechanism is much faster than the 802.1D equivalent
- In just a few seconds, or a small multiple of hello times, most of the entries in the CAM tables of the entire network (VLAN) flush.
- *Why does RSTP not consider link failure a topology change? Because loss of connectivity does not provide new paths*
  - `IF` a switch loses the link to a downstream switch `THEN` the downstream switch:
    1) `IF` the downstream switch has no alternate path `THEN` no action will be taken to improve convergence
    2) `IF` the downstream switch has an alternate path, the downstream switch will unblock it and consequently generate its own BPDUs with the TC bit set
  - Like with STP, PortFast-enabled ports do not create topology changes.

Topology Change Notification in RSTP

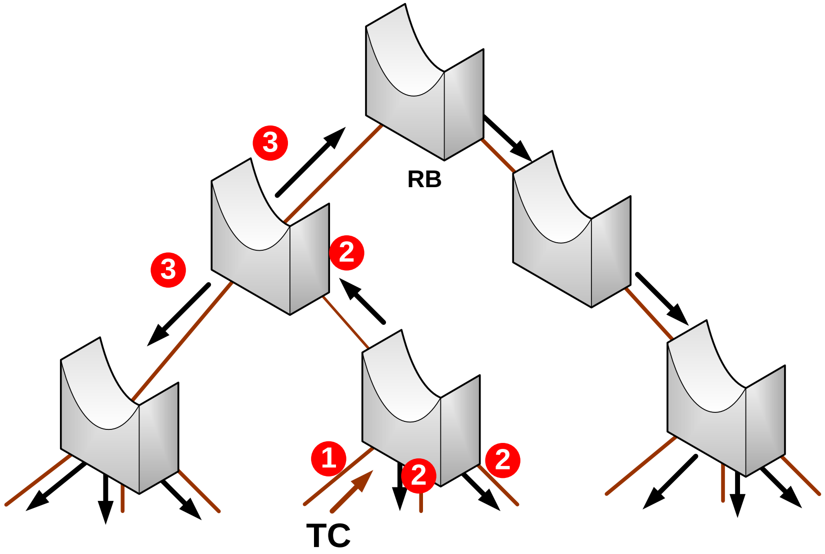
1) TCN is generated only when Non-Edge port transits to Forwarding state

2) Affected switch starts TC-While timer (2× Hello) on each Non-Edge Designated

   - It removes all MAC addresses learned on that ports from CAM table except the port which transits to Forwarding state
   - Switch sends BPDU with TC flag set on until TC-While times out

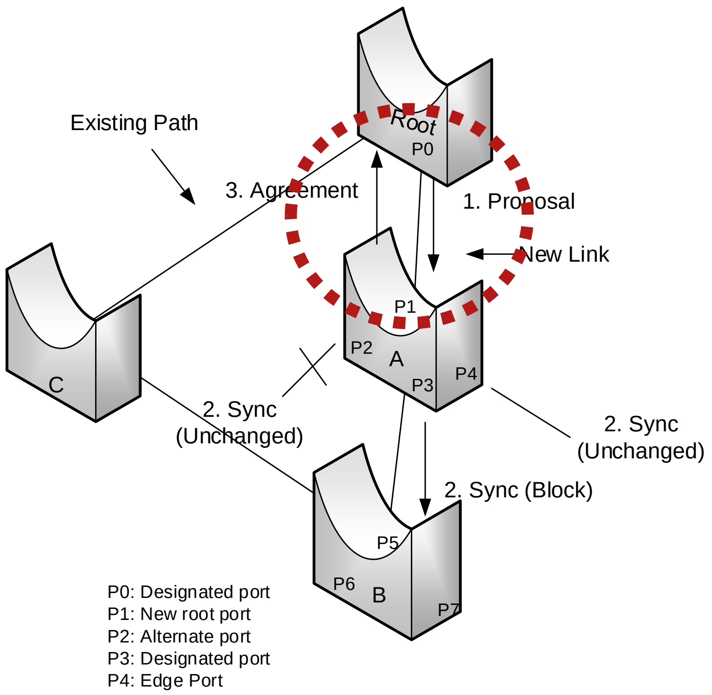
3) Switch which receives BPDU with TC flag

   - Removes all MAC addresses learned on all Non-Edge ports except the one on which it received TC BPDU from CAM table
   - It starts TC-While timer on all Non-Edge ports except the one on which it received TC BPDU
   - Switch sends BPDU with TC flag set on until TC-While times out

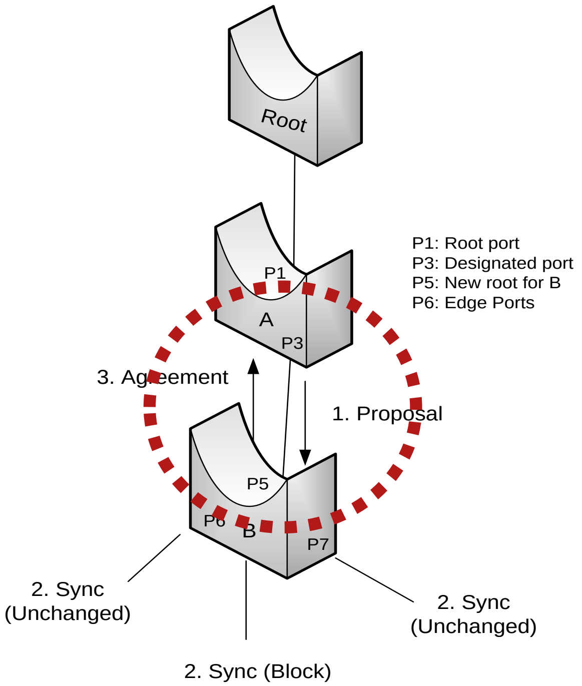


- Changes on Edge ports are not considered as topology change – TC BPDU are not sent and MAC addresses are not removed

RSTP Proposal/Agreement Mechanism ①



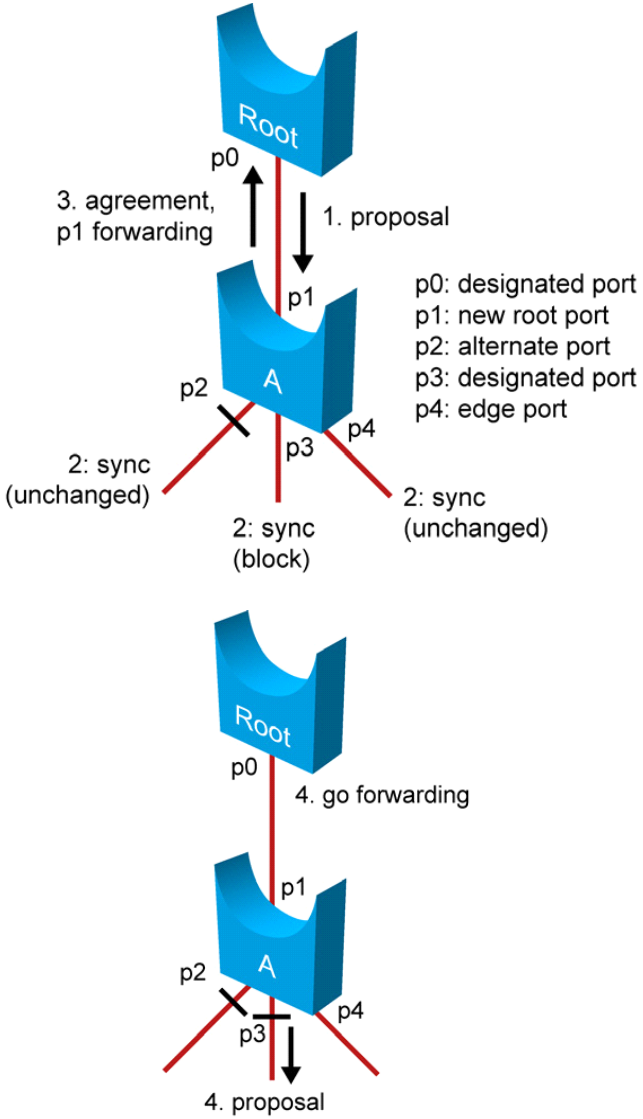
P0: Designated port
P1: New root port
P2: Alternate port
P3: Designated port
P4: Edge Port



P1: Root port
P3: Designated port
P5: New root for B
P6: Edge Ports

- Ports negotiate locally as soon as different BPDUs are received – new link establishment
- Transition occurs as soon as negotiation is completed and then negotiation is then immediately started on other ports
- Usable only on point-to-point links
- Sync state – all Non-Edge Designated ports transit to Discarding state

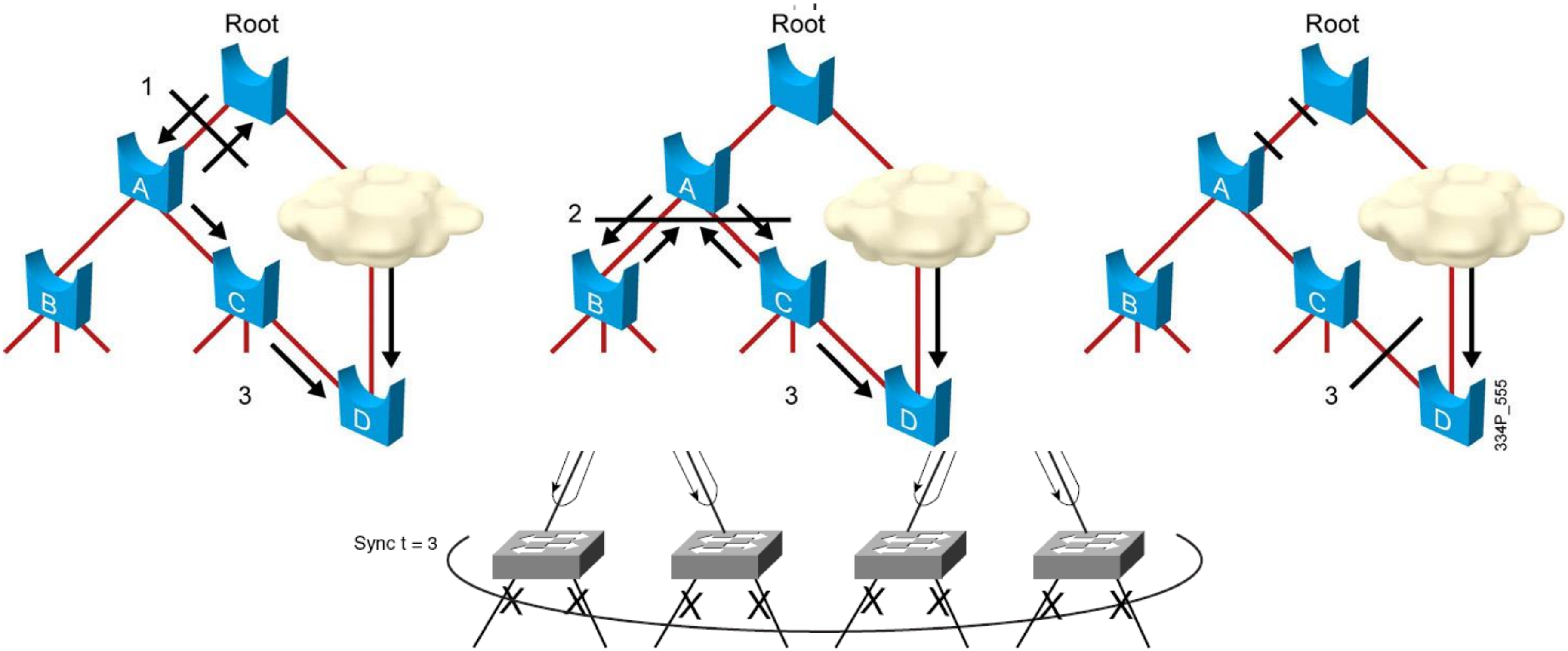
RSTP Proposal/Agreement Mechanism ②



- http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/24062-146.html#agree
- Initial state of port on come up link in RSTP is Designated Discarding
- Each Designated Discarding/Learning port sends BPDUs with Proposal flag set on
- `IF` port on adjacent switch receives Proposal and is going to be Root port `THEN`
  1) Adjacent switch transit all its Non-Edge Designated ports to Discarding state (Sync state)
  2) It sends BPDU with Agreement flag set back and transits outgoing port to Forwarding state
  3) After receiving Agreement initial switch transits own port to Forwarding state
- Designated ports are in Discarding state on adjacent switch as consequence of Sync operation
  - Proposal/Agreement mechanism cascade deeper into topology

Sync Spreading in Topology

- Because RSTP relies on local negotiations, transition occurs as soon as the negotiation is completed



Configuring RSTP

Enabling Rapid STP (RPVST+)

```
! Enabling Rapid PVST+
Switch(config)# spanning-tree mode rapid-pvst
```

```
! Reverting back to PVST+
Switch(config)# spanning-tree mode pvst
```

- Setting priority

```
Switch(config)# spanning-tree vlan VLAN_ID priority NUMBER
Switch(config)# spanning-tree root primary
Switch(config)# spanning-tree root secondary
```

- Show commands

```
Switch# show spanning-tree
Switch# show spanning-tree vlan VLAN_ID
Switch# show spanning-tree detail
Switch# show spanning-tree active
Switch# show spanning-tree root
Switch# show spanning-tree bridge
```

- Debug commands

```
Switch# debug spanning-tree
Switch# debug spanning-tree pvst+
```

Verifying RSTP

```
Switch# show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp

  Root ID     Priority     32768
              Address      0c00c8110000
              Cost         19
              Port         1 (FastEthernet0/1)
              Hello Time   2 sec   Max Age 20 sec   Forward Delay 15 sec
  Bridge ID   Priority     32768
              Address      0c00c8111111
              Hello Time   2 sec   Max Age 20 sec   Forward Delay 15 sec
              Aging Time   300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------------------
Fa0/7               Desg FWD 19        128.9    P2p
Fa0/8               Desg FWD 19        128.10   P2p Peer(STP)
Fa0/9               Desg FWD 19        128.11   P2p
```

Point 2 Point RPVST port

Port backward compatible with STP, neighbor switch is running Legacy STP

Configuring Edge Ports and P2P Links

```
! Edge ports are configured just same as PortFast ports

! Either on global scope...
Switch(config)# spanning-tree portfast default

! ...or per-interface basis
Switch(config)# int range fa 0/1 - 10
Switch(config-if)# spanning-tree portfast
```

```
! Switch automaticaly decides whether link is P2P or not
! based on duplexness of port (P2P ~ full duplex)

! Setting link type manually
Switch(config)# int fa 0/1
Switch(config-if)# spanning-tree link-type point-to-point
```

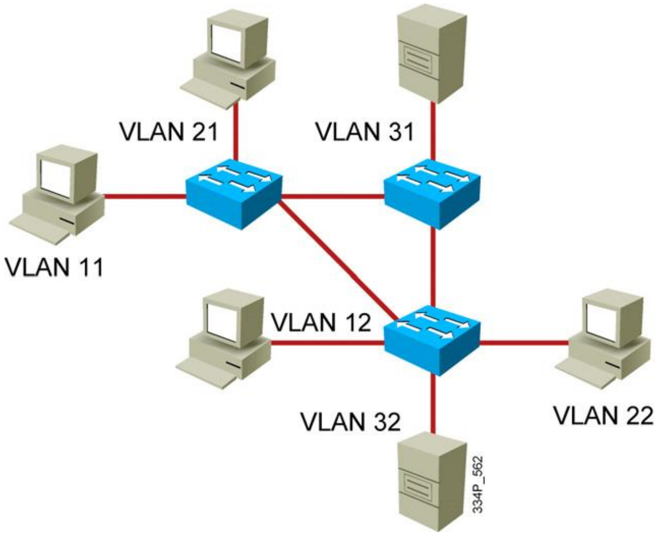
Final Notes on RSTP

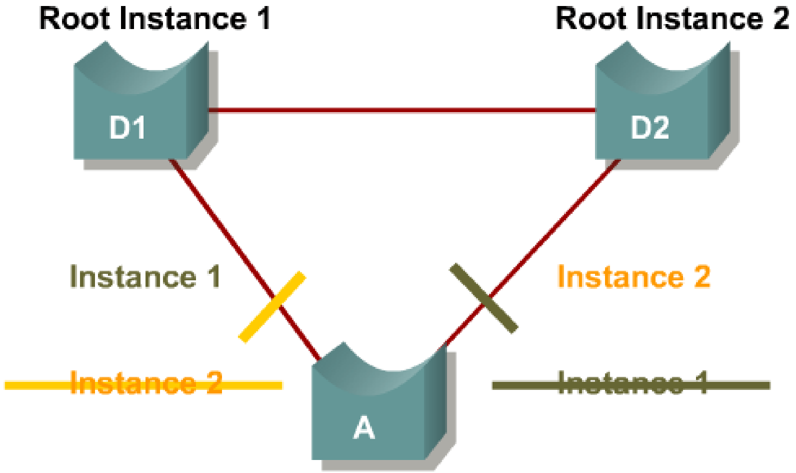
- There is no way how to configure UplinkFast and BackboneFast because they are already part of RSTP
  - UplinkFast idea ~ Alternate port
  - BackboneFast idea ~ RSTP accepts inferior BPDU received by Designated neighbor without waiting for MaxAge expiration
- It is important to mark edge ports with **`spanning-tree portfast`** command in RSTP
  - `IF` NOT whenever there is topological change `THEN` port goes into Sync (Designated Discarding) as and consequence of Proposal/Agreement Mechanism
  - Hence end host risk 30 seconds failures

Multiple STP

Multiple Spanning Tree Protocol

- In some scenarios, many VLANs are spanning several switches. Problem is how many STP instances has to exist:
  - 802.1D: 1 STP for all VLANs
  - Cisco (R)PVST+: 1 STP for each VLAN
  - *Both approaches are not optimal*
- Main idea of MSTP
  - Multiple instances of STP in network could exist but not necessarily with any connection to VLANs
  - Consequently map VLANs onto instances
  - Multiple VLAN could share same instance – *they have same spanning-tree*
  - *Basically grouping instances simplifies the tree structure!!!*
- MSTP is based on RSTP – it has all RSTP advantages
- Cisco Document ID: 24248 "Understanding Multiple Spanning Tree Protocol (802.1s)"
- INE.com "Understanding MSTP"





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

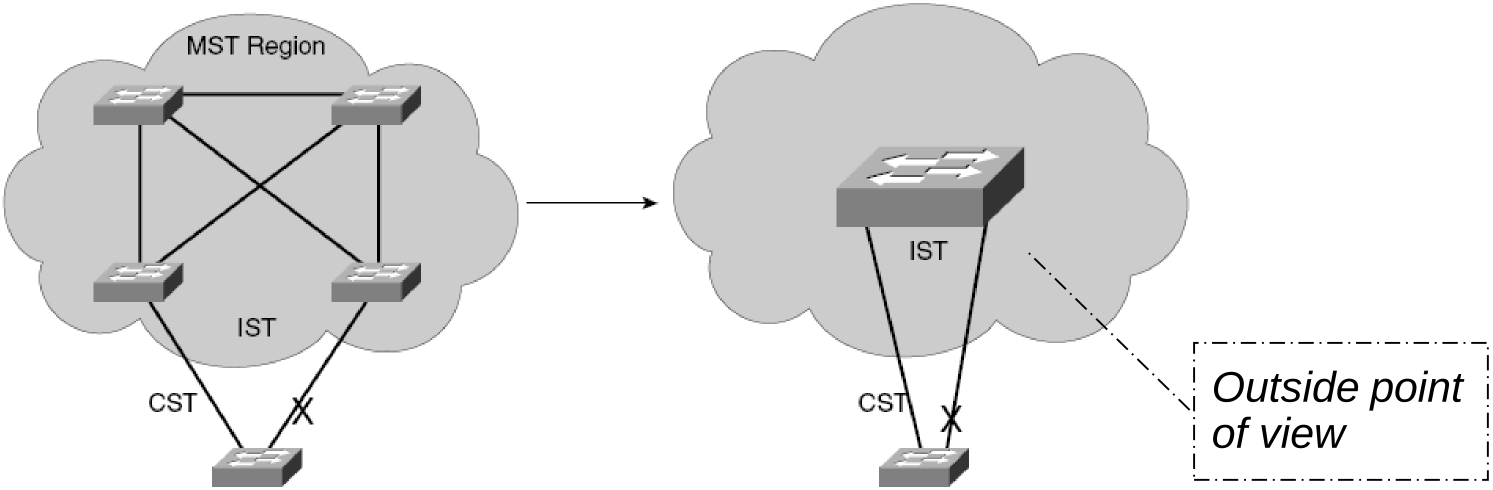
MSTP Region

- When configuring MSTP we must chose how many MSTP instances shall run and which VLANs will be associated with them
- Configuration consists of following components
  - (Text) Name of MST region (32 B)
  - Revision number (2 B)
  - Instances list and VLAN association
    - Mapping is not sent via BPDU, only hash from it
- Switches with identical MST configuration create one region
  - From outside point of view region seems as one undivided switch entity which serves purpose of backward compatibility with (R)STP switches and communication with other regions
  - Borders between regions are on ports receiving BPDU from different region or STP/RSTP/PVST/RPVST BPDU

MSTP Tree Terms

- MSTP establishes confusing terminology referring to different kinds of tree structure it uses
  - **MST Instance (MSTI)**
    - Spanning tree of single MSTP instance
    - MSTI process runs inside region and never reaches behind borders
  - **Internal Spanning Tree (IST)**
    - Spanning tree of MSTI number 0
    - IST instances carries MSTP BPDU and cooperates in interaction with switches outside current MSTP region
  - **Common Spanning Tree (CST)**
    - Spanning tree interconnecting different regions
    - For CST is intra-region spanning tree topology hidden
    - CST sees each region as one giant switch
  - **Common and Internal Spanning Tree (CIST)**
    - Spanning tree of whole switch network in detail – combination of CST and IST

Cooperation between MSTP and STP Switch ①

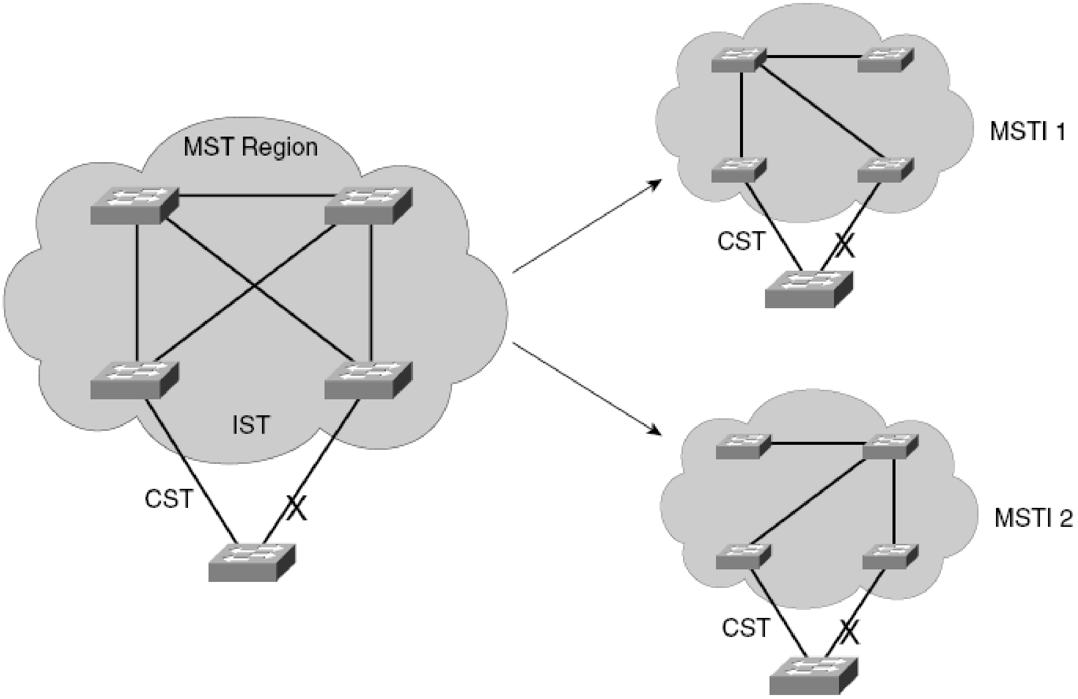


Outside point of view

- MSTP region acts as one big switch
  - `IF` loop is detected `THEN` not just internal ports are blocked but also ports on the borders of region
- Only IST cooperate with “outside world”
  - Other MSTI remains hidden inside region
  - `IF` border port is blocked for IST `THEN` it is blocked for all instances

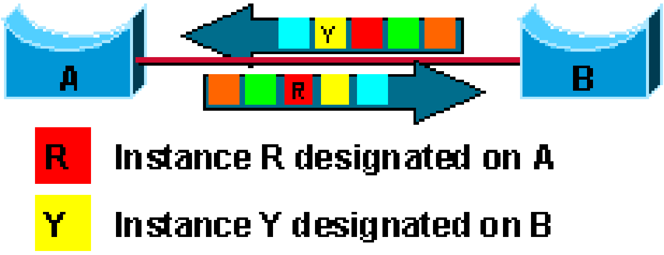
Cooperation between MSTP and STP Switch ②

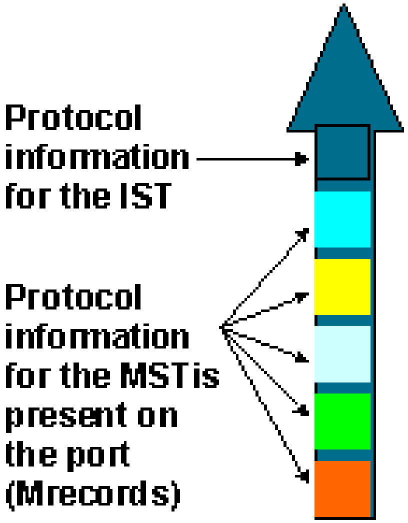
- Internal MSTI has own independent topologies
- On the borders MSTI are controlled by IST state
- IST transfers CST information without any temper inside region
    - Does not increment External RPC
    - Does not modifiy Message Age
    - *Just tunnels external information and this is reason why from outside region seems as one big switch*

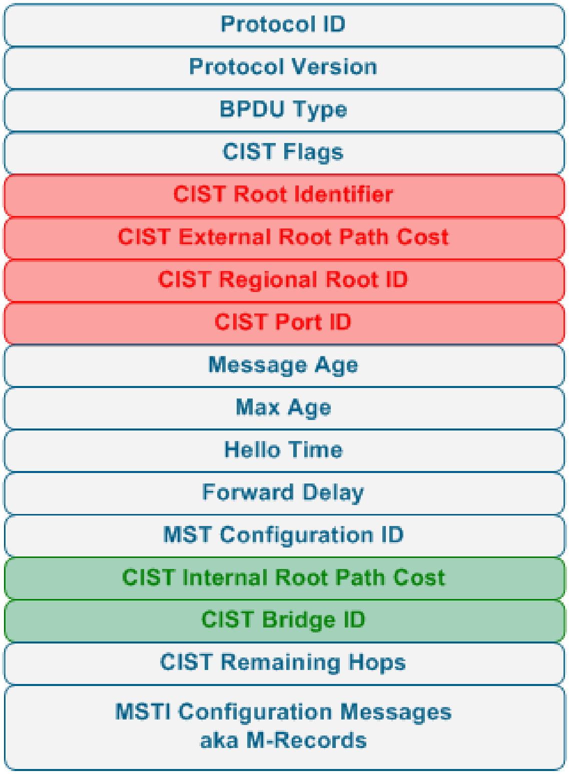


MSTP BPDU

- MSTP sends just one BPDU with all necessary information about IST and all MSTIs
  - **M-record**: Information about one MSTI
  - Timers of MSTI are inherited from IST




Protocol information for the IST
Protocol information for the MSTIs present on the port (Mrecords)



Protocol ID

Protocol Version

BPDU Type

CIST Flags

CIST Root Identifier

CIST External Root Path Cost

CIST Regional Root ID

CIST Port ID

Message Age

Max Age

Hello Time

Forward Delay

MST Configuration ID

CIST Internal Root Path Cost

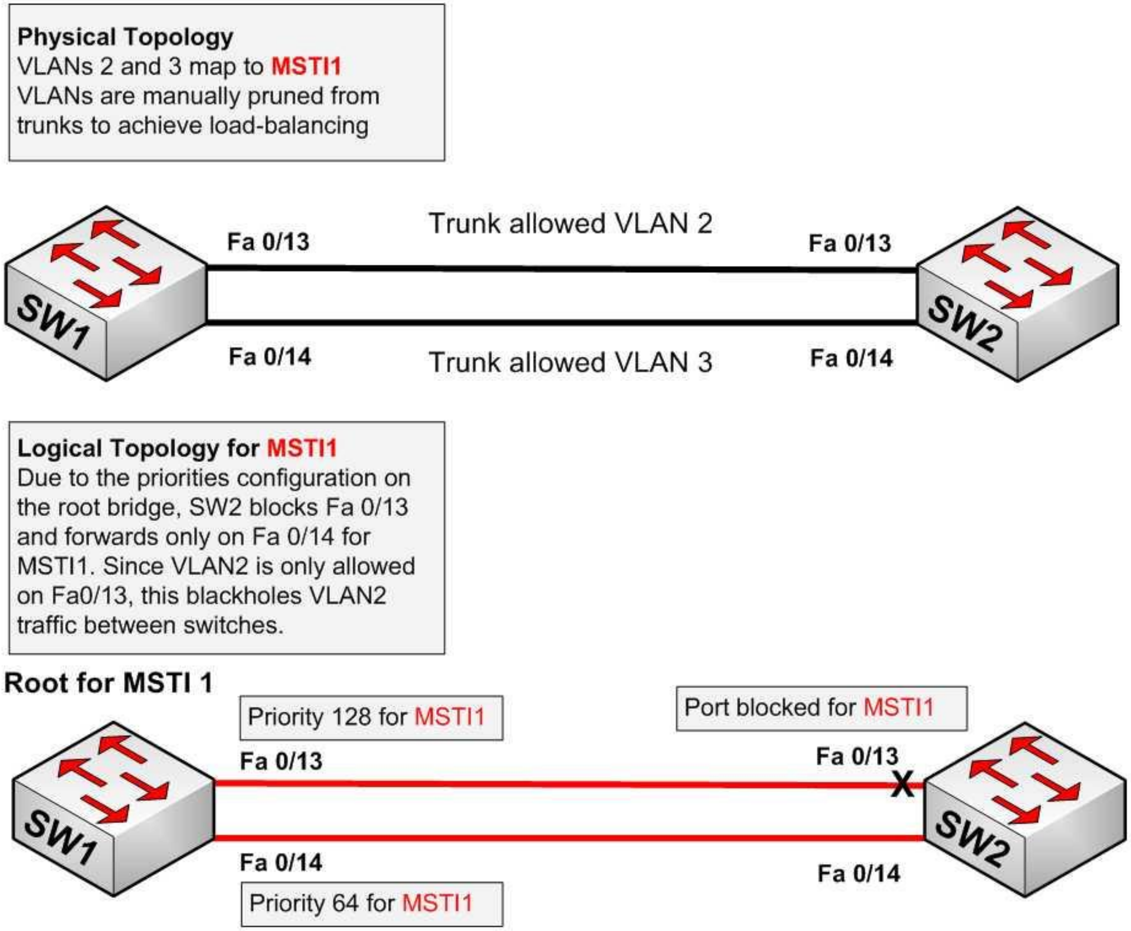
CIST Bridge ID

CIST Remaining Hops

MSTI Configuration Messages aka M-Records

Caveats in MSTP Design ①

- Because MSTP is not tied with VLANs following could happen:



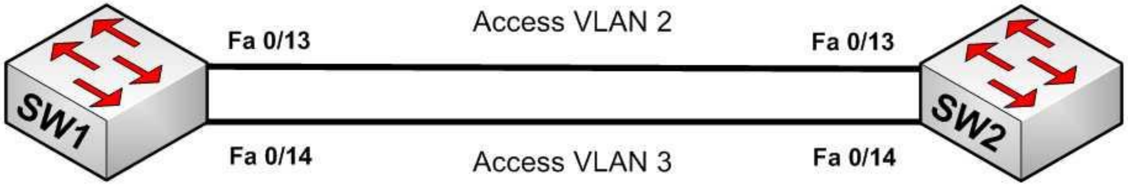
Physical Topology
VLANs 2 and 3 map to **MSTI1**
VLANs are manually pruned from trunks to achieve load-balancing

Logical Topology for MSTI1
Due to the priorities configuration on the root bridge, SW2 blocks Fa 0/13 and forwards only on Fa 0/14 for MSTI1. Since VLAN2 is only allowed on Fa0/13, this blackholes VLAN2 traffic between switches.

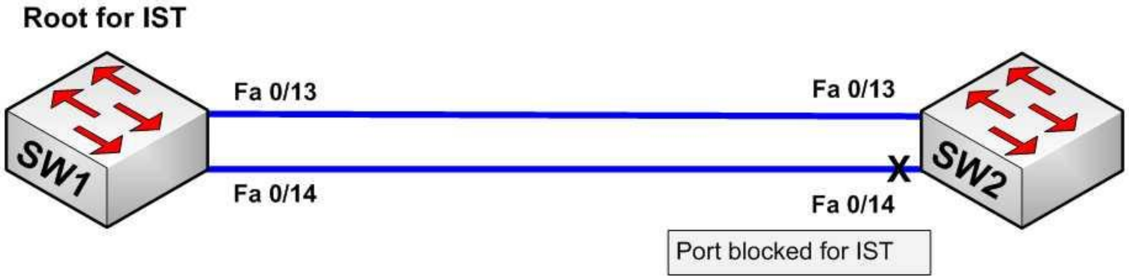
Caveats in MSTP Design ②

- Because MSTP is not tied with VLANs following could happen:

Physical Topology
VLANs 2 and 3 map to default IST (instance 0). Links between switches are non-trunking access ports

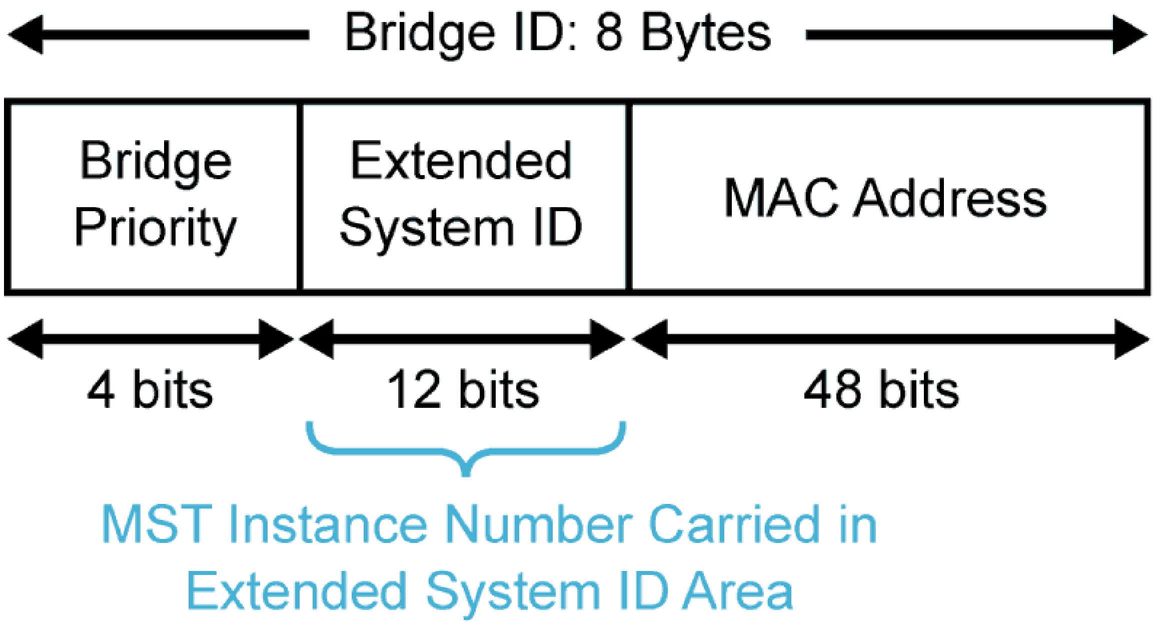


Logical Topology for IST
IST is active on all links. Since there are redundant connections between switches, IST blocks alternate port on SW1. Unfortunately, this is port for VLAN3, and since both VLANs map to IST, VLAN3 traffic is disrupted



Extended System ID

- Used for Bridge identification for MSTP processes
- 12 bit long Extended ID carries MSTP instance number



Configuring MSTP

Configuring MSTP

```
! Enable MST on switch
Switch(config)# spanning-tree mode mst
```

```
! Enter MST configuration
Switch(config)# spanning-tree mst configuration
```

```
! Assigning name to region (max 32 characters)
Switch(config-mst)# name NAME
```

```
! Revision number to track changes
! Whenever reconfiguration occur it is suitable to change
! also revision number on all switches in region
Switch(config-mst)# revision VERSION
```

```
! Association VLAN to MST instance
! By default all VLANs are mapped to IST
Switch(config-mst)# instance INSTANCE_ID vlan VLAN-LIST
```

```
! Displaying current resp. future config
Switch(config-mst)# show { current | pending }
Switch(config-mst)# exit
```

MSTP Configuration Example

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 10-20
Switch(config-mst)# name region1
Switch(config-mst)# revision 1
Switch(config-mst)# show pending
Pending MST configuration
Name [region1]
Revision 1
Instance  Vlans Mapped
-------- ---------------------
0         1-9,21-4094
1         10-20
-------------------------------


Switch(config-mst)# exit
```

Changing Cost, Priority and Timer Values

```
! Switch priority for target instance
Switch(config)# spanning-tree mst INSTANCE-ID priority PRIORITY

! Configuring switch as root switch for target instance
Switch(config)# spanning-tree mst INSTANCE-ID root primary

! Configuring backup switch for target instance
Switch(config)# spanning-tree mst INSTANCE-ID root secondary

! Per-interface configuration
Switch(config)# interface interface-id
Switch(config-if)#
   spanning-tree mst INSTANCE-ID port-priority PRIORITY
Switch(config-if)# spanning-tree mst INSTANCE-ID cost COST
```

```
! Setting timers for all instances
Switch(config)# spanning-tree mst hello-time seconds
Switch(config)# spanning-tree mst forward-time seconds
Switch(config)# spanning-tree mst max-age seconds
```

Verifying MSTP

```
! MSTP status
Switch# show spanning-tree

! MSTP region configuration
Switch# show spanning-tree mst configuration

! General MSTP information
Switch# show spanning-tree mst

! Information for target instance
Switch# show spanning-tree mst INSTANCE_ID

! Detail information for targer instance
Switch# show spanning-tree mst INSTANCE_ID detail
```

The show spanning-tree Command

```
DLS2# sh spanning-tree

MST0
  Spanning tree enabled protocol mstp
  Root ID     Priority     32768
              Address      0017.9446.ad00
              Cost         0
              Port         13 (FastEthernet0/11)
              Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority     32768  (priority 32768 sys-id-ext 0)
              Address      0017.9460.3080
              Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Fa0/7               Desg FWD 200000    128.9    P2p
Fa0/8               Desg FWD 200000    128.10   P2p
Fa0/9               Desg FWD 200000    128.11   P2p
Fa0/10              Desg FWD 200000    128.12   P2p
Fa0/11              Root FWD 200000    128.13   P2p
Fa0/12              Altn BLK 200000    128.14   P2p



MST1
  Spanning tree enabled protocol mstp
  Root ID     Priority     32769
...
...
```

The sh span mst config Command

```
Switch# sh spanning-tree mst configuration

DLS1#sh span mst configuration
Name      [cisco]
Revision  1     Instances configured 3

Instance  Vlans mapped
--------  -------------------------------------------------------------------
0         1-19,51-79,81-99,101-4094
1         20-50
2         80,100
-----------------------------------------------------------------------------
```

The `sh spanning-tree mst` Command ①

```
Switch# show spanning-tree mst
##### MST0     vlans mapped:   1-19,51-79,81-99,101-4094
Bridge         address 0017.9446.ad00  priority      32768 (32768 sysid 0)
Root           address 0017.9460.3080  priority      1     (0 sysid 1)
               port    Fa0/11          path cost     200000
Regional Root this switch
Operational    hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured     hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/7            Altn BLK 200000    128.9    P2p Bound(PVST)
Fa0/8            Altn BLK 200000    128.10   P2p Bound(PVST)
Fa0/9            Altn BLK 200000    128.11   P2p Bound(PVST)
Fa0/10           Altn BLK 200000    128.12   P2p Bound(PVST)
Fa0/11           Root BKN*200000    128.13   P2p Bound(PVST) *PVST_Inc
Fa0/12           Altn BLK 200000    128.14   P2p Bound(PVST)

##### MST1     vlans mapped:   20-50
Bridge         address 0017.9446.ad00  priority      32769 (32768 sysid 1)
Root           this switch for MST1

Interface        Role Sts Cost      Prio.Nbr Type
 --More--
```

The sh spanning-tree mst Command ②

```
DLS2(config)# do sh spanning-tree mst 2

##### MST2    vlans mapped:   80,100
Bridge        address 0017.9460.3080  priority      24578 (24576 sysid 2)
Root          this switch for MST2

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/7            Desg FWD 200000    128.9    P2p
Fa0/8            Desg FWD 200000    128.10   P2p
Fa0/9            Desg FWD 200000    128.11   P2p
Fa0/10           Desg FWD 200000    128.12   P2p
Fa0/11           Desg FWD 200000    128.13   P2p
Fa0/12           Desg BLK 200000    128.14   P2p Dispute
```

The sh span mst int Command ①

```
DLS1# sh spanning-tree mst 1 int fa 0/7

FastEthernet0/7 of MST1 is designated forwarding
Edge port: no             (default)        port guard : none        (default)
Link type: point-to-point (auto)           bpdu filter: disable     (default)
Boundary : internal                        bpdu guard : disable     (default)
Bpdus (MRecords) sent 1069, received 6

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-------- ---- --- --------- -------- -------------------------------
1        Desg FWD 200000    128.9    20-50
```

The sh span mst detail Command

```
DLS1# sh spanning-tree mst 1 detail

##### MST1    vlans mapped:   20-50
Bridge        address 0017.9446.ad00  priority      32769 (32768 sysid 1)
Root          this switch for MST1

FastEthernet0/7 of MST1 is designated forwarding
Port info             port id          128.9  priority    128  cost      200000
Designated root       address 0017.9446.ad00  priority  32769  cost           0
Designated bridge     address 0017.9446.ad00  priority  32769  port id    128.9
Timers: message expires in 0 sec, forward delay 0, forward transitions 3
Bpdus (MRecords) sent 1101, received 6

FastEthernet0/8 of MST1 is designated forwarding
Port info             port id         128.10  priority    128  cost      200000
Designated root       address 0017.9446.ad00  priority  32769  cost           0
Designated bridge     address 0017.9446.ad00  priority  32769  port id   128.10
Timers: message expires in 0 sec, forward delay 0, forward transitions 1
Bpdus (MRecords) sent 1069, received 4

FastEthernet0/9 of MST1 is designated forwarding
Port info             port id         128.11  priority    128  cost      200000
Designated root       address 0017.9446.ad00  priority  32769  cost           0
Designated bridge     address 0017.9446.ad00  priority  32769  port id   128.11
 --More--
```

VTPv3 and MSTP Configuration

- VTPv3 allows to synchronize MSTP region
  - One VTP domain must be identical to one MSTP region
  - All switches must support VTPv3
- Configuration changes to MSTP region should be issued only on primary VTP server
  - Configuration will be delivered via VTP to other switches
- VTPv3 eases management of MSTP in switched network and also reduces occasional failures during configuration migration
  - Beware of switch with different region configuration than its neighbors – switch itself creates own region and port blocking could occur on switches border ports
  - When using manual configuration previous accident could happen even more
  - Hence using VTPv3 minimizes existence of isolated regions in network

VTPv3 Configuration Example

```
! On each switch in VTP domain it is necessary to issue following
Switch(config)# vtp version 3
Switch(config)# vtp domain NAME
Switch(config)# vtp mode server mst
! Or instead of the last command could be used also
Switch(config)# vtp mode client mst


! On VTP server switch which suppose to be primary server for MSTP
! database it must be issued command
Switch# vtp primary mst

! Only after previous command we can start configuring MSTP.
! Each MSTP region configuration change is then transferred via VTP
```

Protecting STP

Protecting STP

- Spanning-Tree Toolkit (SPT Toolkit)
- On Cisco devices is present variety of features intended to protect STP consistence in many ill-behaved scenarios
  - **BPDU Guard**
  - **BPDU Filter**
  - **Root Guard**
  - **Etherchannel Misconfig Guard**
  - **Loop Guard**
- „Troubleshooting Spanning Tree PVID- and Type-Inconsistencies“

STP BPDU Guard

- **BPDU Guard** transits port to err-disabled state upon reception of any BPDU
  - Protection on access port – they are connecting end hosts so there is no reason why other switch should be connected to them
  - Majorly important for PortFast ports which tends to unblock them as standard STP port upon reception of BPDU
- Configuration snippet (global vs. per-interface)

```
! Global configuration
Switch(config)# spanning-tree portfast bpduguard default
! Per-interface configuration
Switch(config)# int fa0/1
Switch(config-if)# spanning-tree bpduguard enable
```

- „Spanning-Tree Protocol Enhancements using Loop Guard and BPDU Skew Detection Features“

STP BPDU Filter ①

- *Sometimes it is desirable when switch port is not sending BPDU messages*
  - They are completely unnecessary for end host and could cause security threat
  - Sometimes network should be split into two independent STP domains (e.g. during migration)
- **BPDU Filter** serves this purpose and filters sending of BPDUs
- Its behavior depends whether it is configured globally or per-interface
- When BPDU Filter is activated on global scope
  - It is working in PortFast ports
  - Whenever port goes up it initially sends 10 – 11 BPDU
    - `IF` port does not receive any BPDU upon initial sending this `THEN` port stops generating BPDUs
    - `IF` port receives any BPDU `THEN` BPDU Filter and PortFast is deactivated on port and it becomes ordinary STP port

STP BPDU Filter ②

- When BPDU Filter is activated per-interface basis
  - Port does not send BPDUs
  - Received BPDUs are discarded
- Configuration snippet:

```
! Global configuration
Switch(config)# spanning-tree portfast bpdufilter default
! Per-interface configuration
Switch(config)# int fa0/1
Switch(config-if)# spanning-tree bpdufilter enable
```

STP Etherchannel Misconfig Guard

- STP sees Etherchannel as one logical connection
  - Physically Etherchannel consists of multiple ports
  - But BPDUs are sent only through the single of all Etherchannel ports – in **`show etherchannel summary`** command it is marked as “default”
- Previous fact is good indicator whether ports on both ends of link are bonded into Etherchannel
  - `IF` BPDUs are received on more than one port `THEN` ports on opposite ends are not properly bonded in Etherchannel
- This check is done by **Etherchannel Misconfig Guard**
  - `IF` error is detected `THEN` port transits to err-disabled state
- It is enabled on global scope by following command

```
Switch(config)# [no] spanning-tree etherchannel guard misconfig
```

- „Understanding EtherChannel Inconsistency Detection“

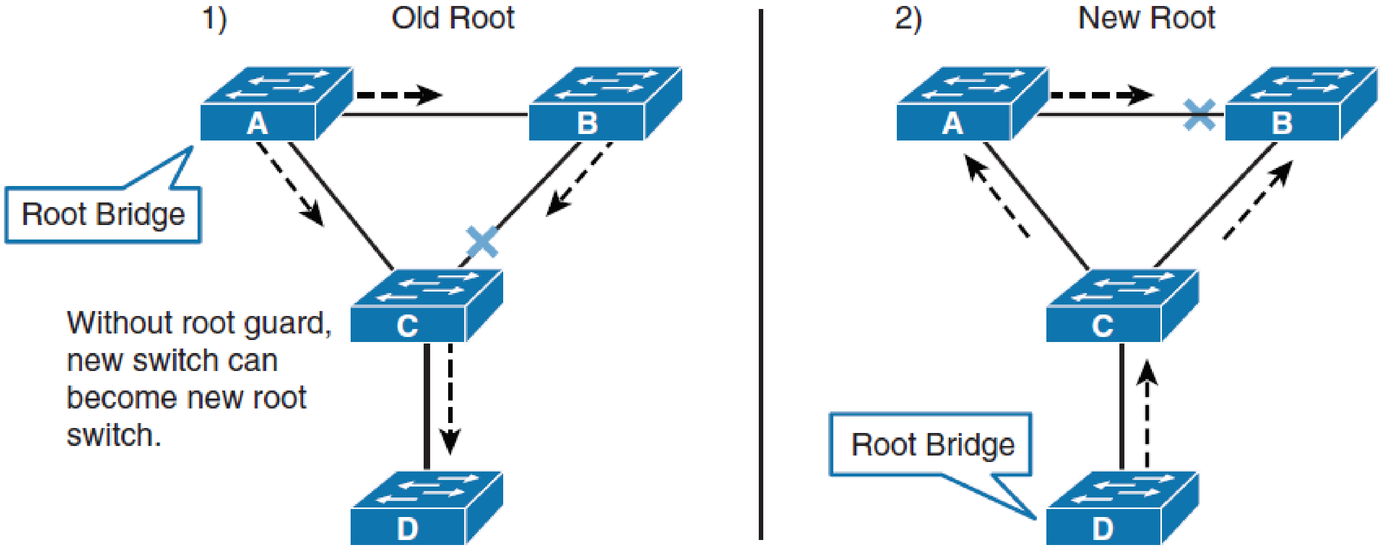
STP Root Guard

- *When we are merging our network with customer’s to the one STP domain we need to protect our root bridge*
  - Other than intended root bridge could be elected accidently or by inappropriate manipulation with metrics
- **Root Guard** (configured on per-interface basis) prevents switching from Designated to Root port role
  - `IF` superior BPDU is received on port that should change its role to Root port `THEN` port transits to root-inconsistent blocking state
  - Root-inconsistent state is automatically turned off in 20 seconds after reception of last superior BPDU enforcing improper path to root bridge
  - Hence, Root bridge functionality in network is guaranteed and secured
- Root Guard is configured on individual port

```
Switch(config)# int fa0/1
Switch(config-if)# spanning-tree guard root
```

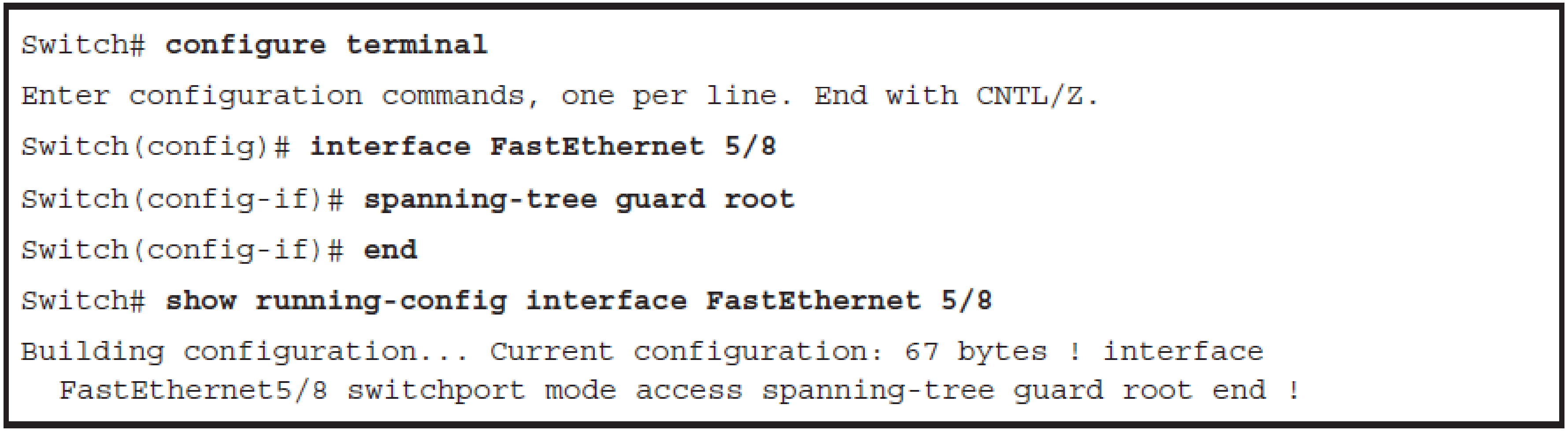
- „Spanning Tree Protocol Root Guard Enhancement“

Use Root Guard

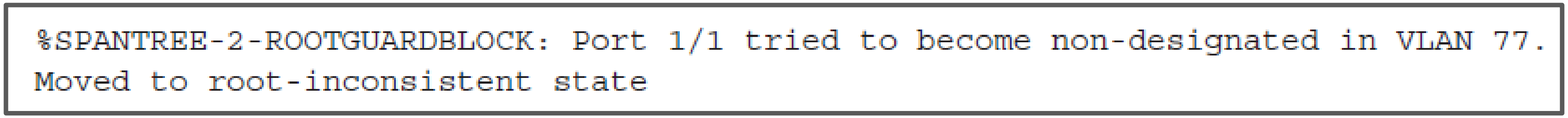


- The current design recommendation is to enable Root Guard on all access ports so that a root bridge is not established through these ports.

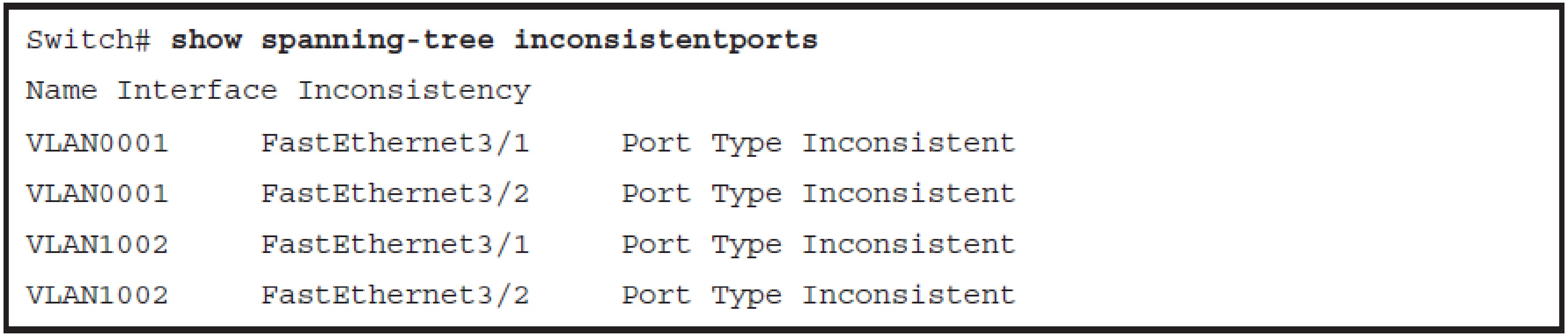
Configuring and Verifying Root Guard



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

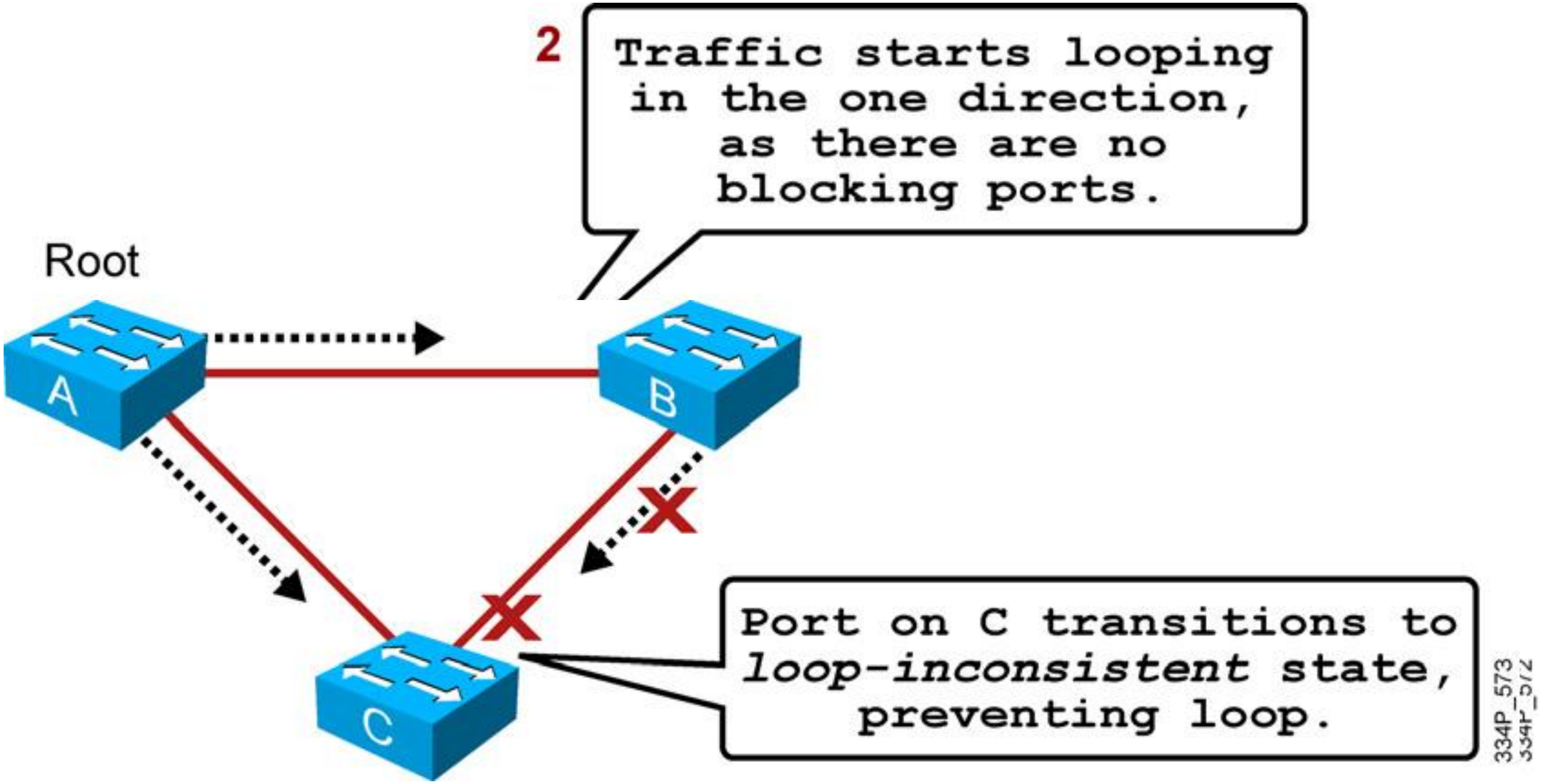


```
%SPANTREE-2-ROOTGUARDBLOCK: Port 1/1 tried to become non-designated in VLAN 77.
Moved to root-inconsistent state
```



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

Loop Guard Motivation



Traffic starts looping in the one direction, as there are no blocking ports.

Port on C transitions to *loop-inconsistent* state, preventing loop.

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

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

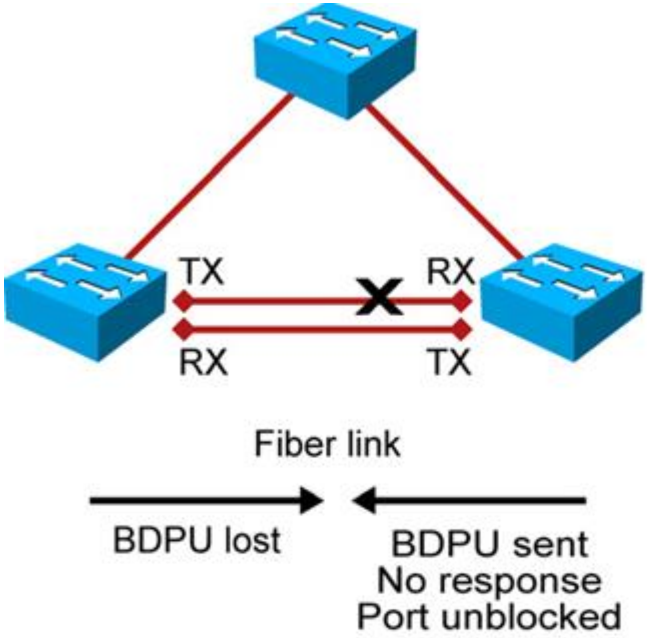
STP Loop Guard

- Loop Guard main idea is based on fact that port which suppose to be blocked must constantly receive superior BPDUs
  - `IF` Blocking port stops to receive superior BPDUs `THEN` Loop Guard transits this port to loop-inconsistent blocking state
  - Loop-inconsistent state is removed automatically immediately upon reception of superior BPDU
- **Loop Guard** prevents Root and Alternate ports become Designated ports
- Loop Guard is configured globally or individually on interfaces
  - Globally enabled Loop Guard is activated only on STP Point-to-Point link types

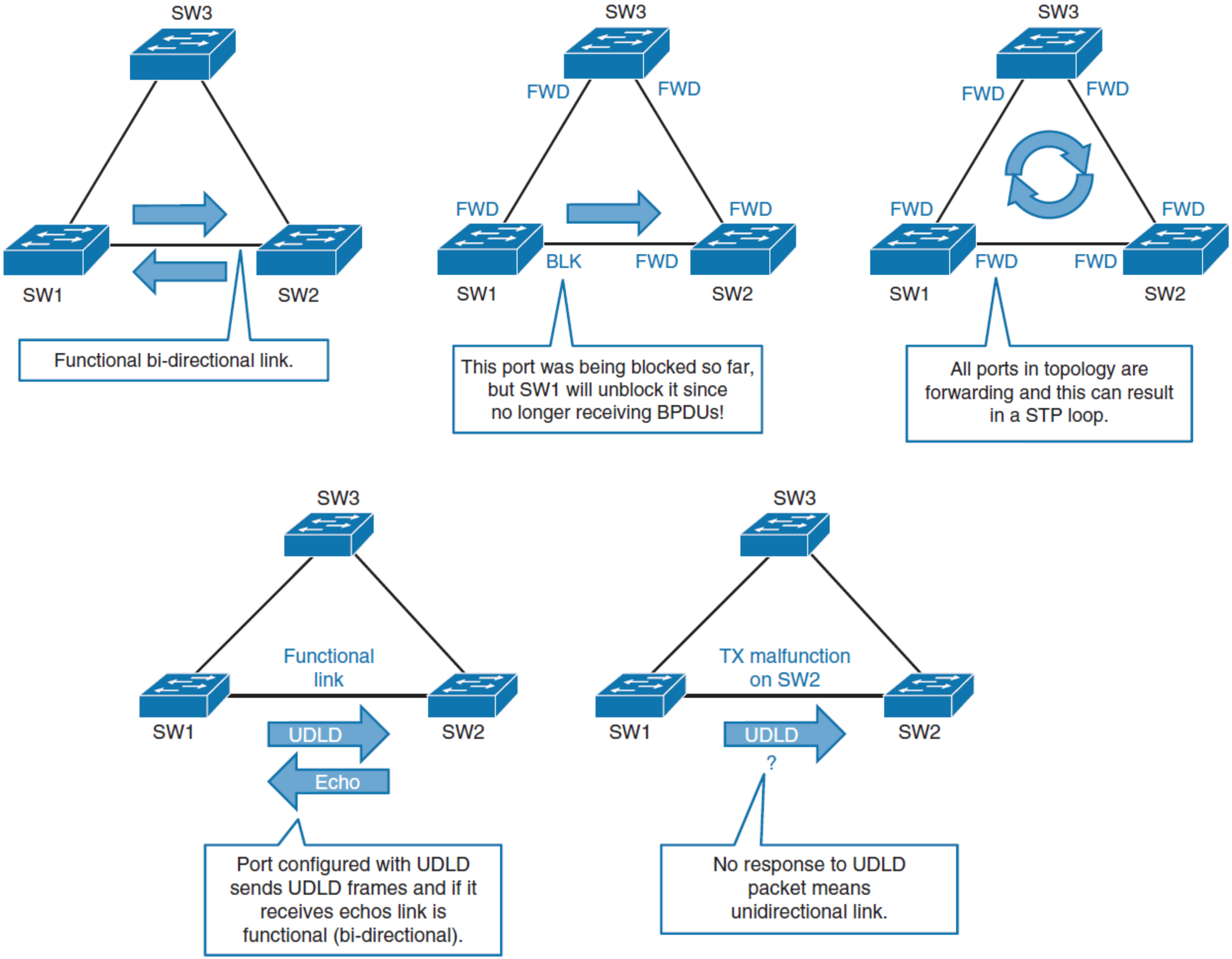
```
! Global configuration
Switch(config)# spanning-tree loopguard default
! Per-interface configuration
Switch(config)# int fa0/1
Switch(config-if)# spanning-tree guard loop
```

UniDirectional Link Detection

- *STP Loop Guard is nice logic behind STP, but it cannot solve certain failure situation*
  - Optic cable could be wrongly interconnected – Tx is terminated somewhere else than on Rx part on the opposite side
  - Missing BPDUs on Etherchannel deactivates whole Etherchannel
  - STP works above logical port and VLAN, it's granularity is fairly rough
- There exist separate protocol capable of detecting unidirectional physical connections – **UDLD**
  - U.S. Patent 7480251
  - Document ID: 10591, „Understanding and Configuring the Unidirectional Link Detection Protocol Feature"



UDLD Motivation



Functional bi-directional link.

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

All ports in topology are forwarding and this can result in a STP loop.

Port configured with UDLD sends UDLD frames and if it receives echos link is functional (bi-directional).

No response to UDLD packet means unidirectional link.

UDLD Specifics

- UDLD is a Cisco proprietary that is also defined in RFC 5171
- UDLD is keepalive/echo L2 protocol working above physical point-to-point link and L1 mechanism
  - Two devices send each other UDLD packets inside which is echod they own identity
  - `IF` port stops to receive UDLD packet OR neighbor stops sending their own ID `THEN` problem might occurred
- Both UDLD peers discover each other by exchanging special frames that are sent to well-known MAC address 01:00:0C:CC:CC:CC
- UDLD messages are sent at regular intervals. This timer can be modified. The default setting varies between platforms. The typical value is 15 seconds.

UDLD Specifics

- In an EtherChannel bundle, UDLD will error-disable only the physical link that has failed.
- **UDLD Normal Mode**: `IF` port stops receiving UDLD `THEN` it remains in „undetermined“ state and UDLD doesn’t do anything
  - Suitable when L1 has own detecting mechanism for detecting unidirectional link
  - UDLD serves as controlling mechanism whether link is not shared or even connected to same device
- **UDLD Aggressive Mode**: `IF` port stops receiving UDLD `THEN` UDLD deactivates port
  - Suitable also for metallic interconnections

Configuring UDLD

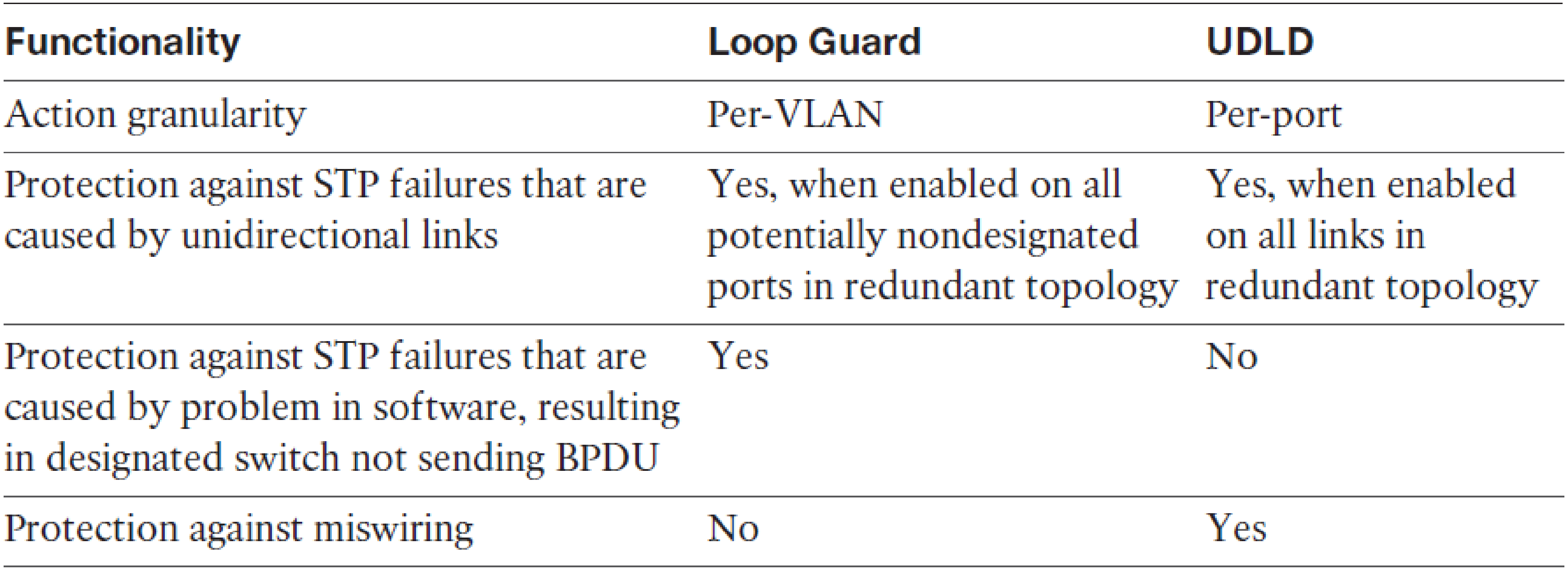
- UDLD could be activated globally or per-interface basis
  - IF UDLD is activated globally THEN it works only optical interfaces (fiber optic)

```
! Global configuration turns it only on optic
Switch(config)# udld { enable | aggressive }
! Per-interface configuration
Switch(config)# int fa0/1
Switch(config-if)# udld port [ aggressive ]
```

- When UDLD deactivated port it could be reactivated issuing following command:

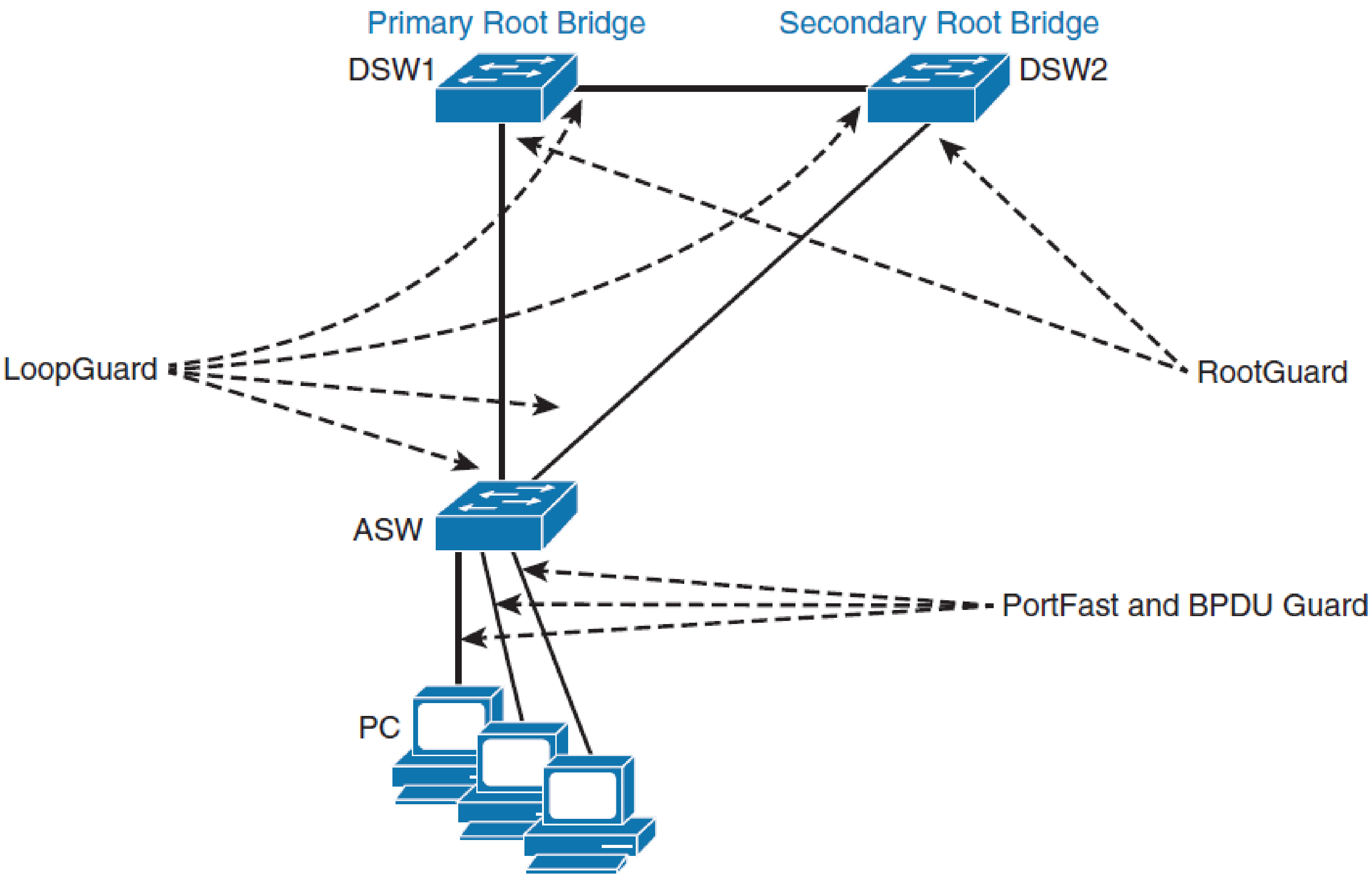
```
Switch# udld reset
```

Comparing LoopGuard and UDLD



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

STP Stability Recommendations



STP Stability Recommendations

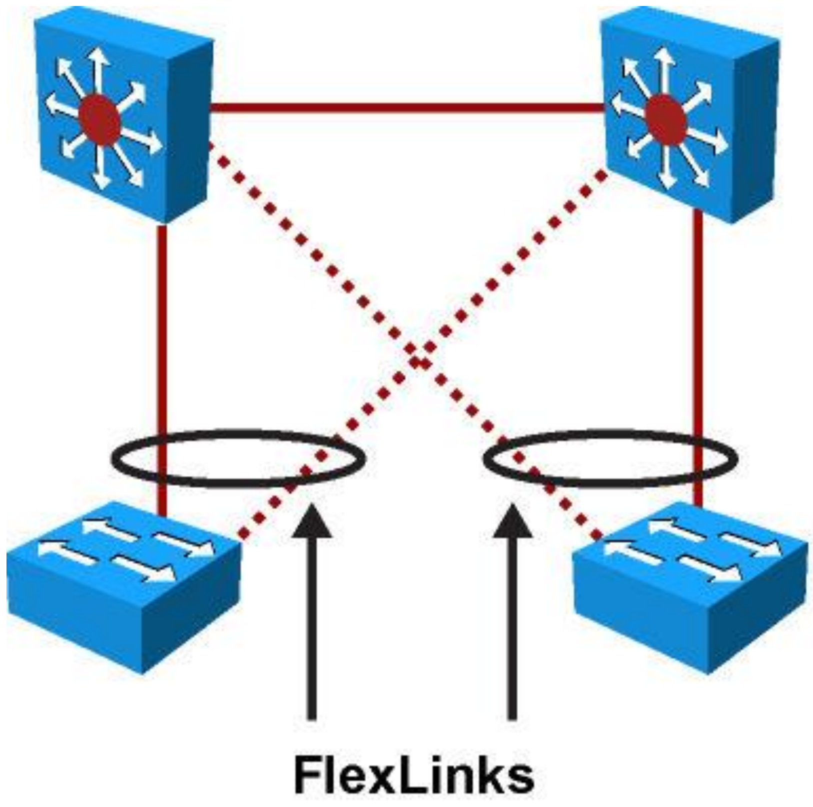
- **PortFast:** Apply to all end-user ports. To secure PortFast-enabled ports, always combine PortFast with **BPDU Guard**.
- **Root Guard:** Apply to all ports where root is never expected.
- **Loop Guard:** Apply to all ports that are or can become nondesignated.
- **UDLD:** The UDLD protocol enables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

Useful Command

- **`show interface status err-disabled`**
- **`show udld interface`** *`IFACE`*

Flex Links

Flex Links



- Flex Links is proprietary solution intended for access switches
- It replaces STP in redundant topology by simply defining interface’s backing up port
  - `IF` active goes down `THEN` backup replaces its functionality
  - Standby interface might relay data for different VLAN for purposes of load-balancing
- Because of more simpler logic Flex Links react more swiftly to topology changes than STP
- On Flex Links interfaces is STP deactivated
  - Hence, Flex Links must be present on all uplinks from access switch
  - Otherwise switching loops might occur

Flex Links Configuration How To

- Flex Link could be configured on physical L2 ports or Port-Channels
- Data are passed only through active interface
  - Standby interface is up but does not relay data
  - It is possible to load-balance between active and standby with appropriate configuration
- Failover is in the 1-to-2-second range, immediately after this standby interface becomes new active interface and starts to transmit data
  - `IF` former active interface goes up `THEN` it stays in backup state – it is non-preemptive by default
  - Preemption could be configured
- Only one interface could be marked as standby for target interface
- Either active or standby interface could be member of only one Flex Links pair
- On Flex Link ports is STP disabled automatically

Configuring and Verifying Flex Links

- Flex Links is configured on active interface referencing backup interface with command **switchport backup interface**

```
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
```

Where to go next?

- Cisco Spanning Tree Protocol Configuration Guide:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/command/reference/3560cr.html

- Configuring MST Configuration Guide:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/swstp.html

- Cisco Optional Spanning-Tree Features Configuration Guide:

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/swmstp.html

  www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_52_se/configuration/guide/swstpopt.html

Slides adapted by Vladimír Veselý partially from official course materials but the most of credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

The last update: 2016-10-17