Spanning Tree Protocol (Common Spanning Tree (CST)). (802.1D)

spanning-tree helps you to create a loop-free topology in your switched network.

Spanning tree protocol (STP) (IEEE 802.1D) is predominantly used **to prevent layer 2 loops and broadcast storms** and is also used **for network redundancy**.

Redundant topology causes **broadcast storms, multiple frame copies, and MAC address table instability problems**.

Switches within the same network need to be enabled for STP before they run the spanning tree algorithm so they can accurately **determine which switch should be elected the "root bridge."** This **root bridge will be responsible for sending bridge protocol data units (BPDUs) along with other information to its directly connected switches that, in turn, forward the BPDUs to their neighboring switches**. Each switch has a **bridge ID (BID),** which is a **combination of a priority value (default 32768) and the switch's own MAC address.**
Root bridge= bridge with the lowest BID.

STP Port States

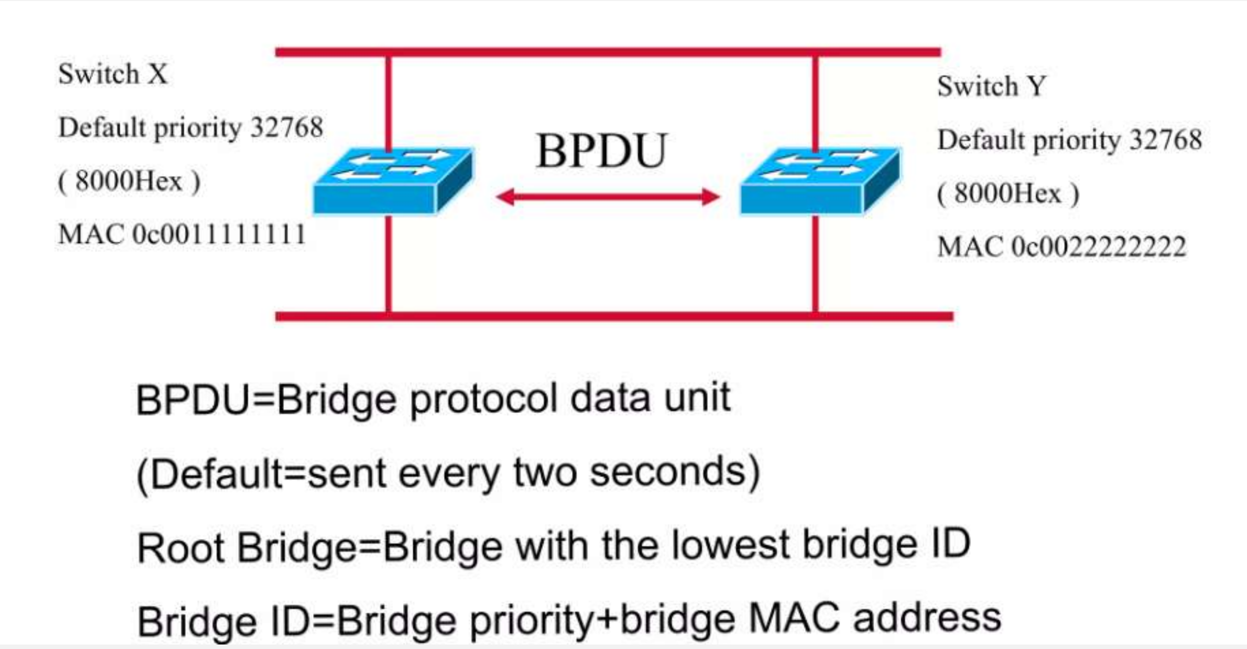
- **Disabled**: - The result of an **administrative command** that will disable the port.
- **Blocking: -** When a device is **connected, the port will first enter the blocking state. (20 sec)**
- **Listening: -** The switch **will listen for and send BPDUs. (15 sec)**
- **Learning: -** The switch **will receive a BPDU,** will stop sending its own BPDUs.**(15 sec)**
- **Forwarding**: - The port **is forwarding** traffic.

STP Port Roles

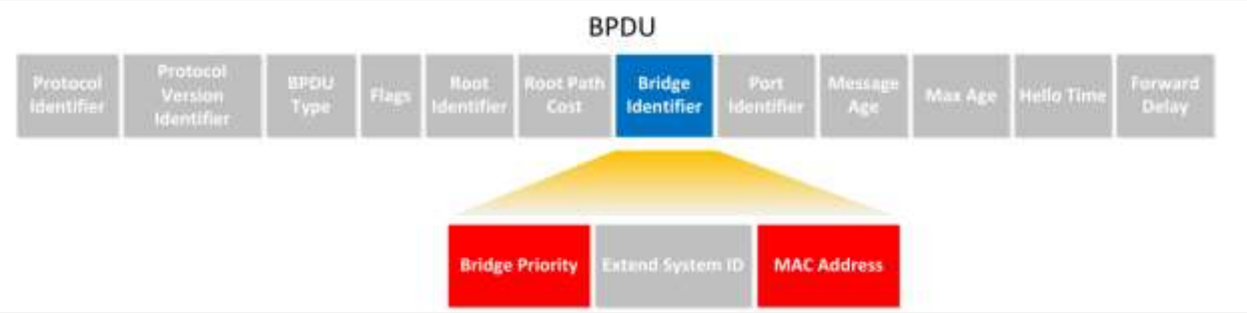
- **Root** - Ports on **non-root switches with the best cost path to root bridge**. These ports forward data to the root bridge.
- **Designated** - Ports on **root switches**. **All ports on the root bridge will be designated.**
- **Blocked** - **All other ports** to bridges or switches are in a **blocked state**.

State	Can forward data?	Learn MAC?	Timer	Transitory or Stable State?
Blocking	No	No	Max Age (20 sec)	Stable
Listening	No	No	Forward Delay (15 sec)	Transitory
Learning	No	Yes	Forward Delay	Transitory
Forwarding	Yes	Yes		Stable

STP Root Bridge Election



Since spanning tree is enabled, all our switches will send a special frame to each other called a **BPDU. "Bridge Protocol Data Unit"**

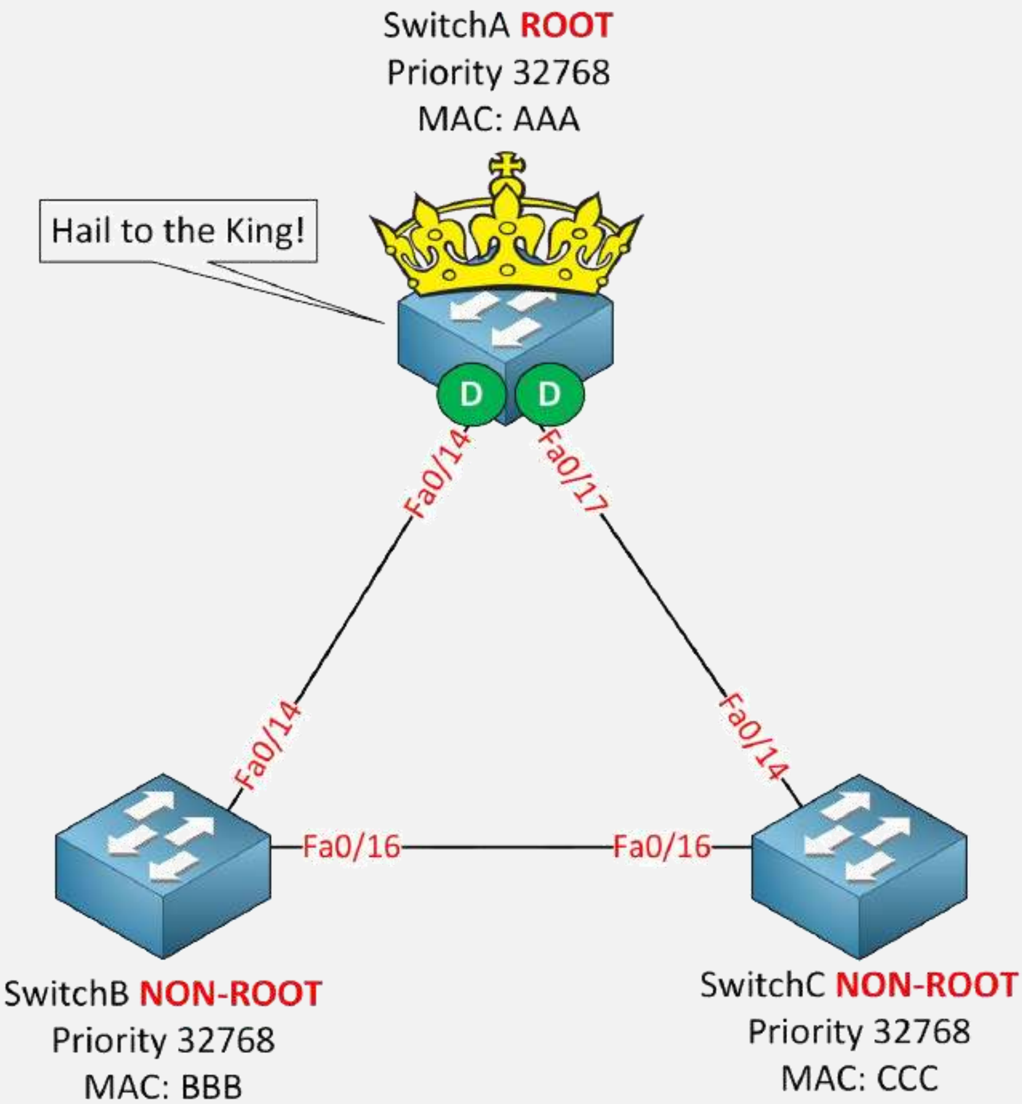


Bridge ID = 8 byte → 6byte for mac address and 2 for bridge priority (by default 32768)

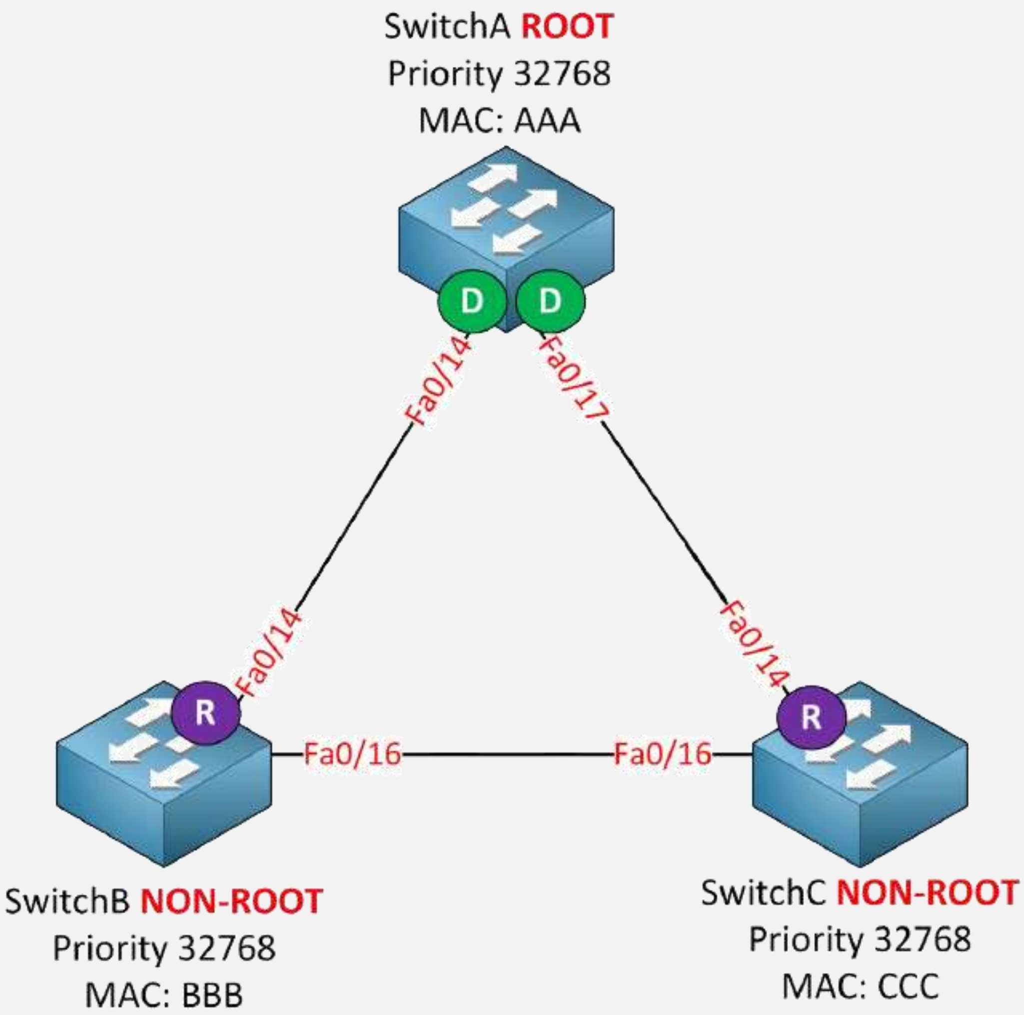
Spanning-Tree Operation

- **One root bridge per network.**
- **One root port per nonroot bridge.**
- **One designated port per segment.**
- **Nondesignated ports are unused.**

The first thing that spanning-tree has to do is **elect a root bridge**. The root bridge **is the switch with the lowest bridge id**.

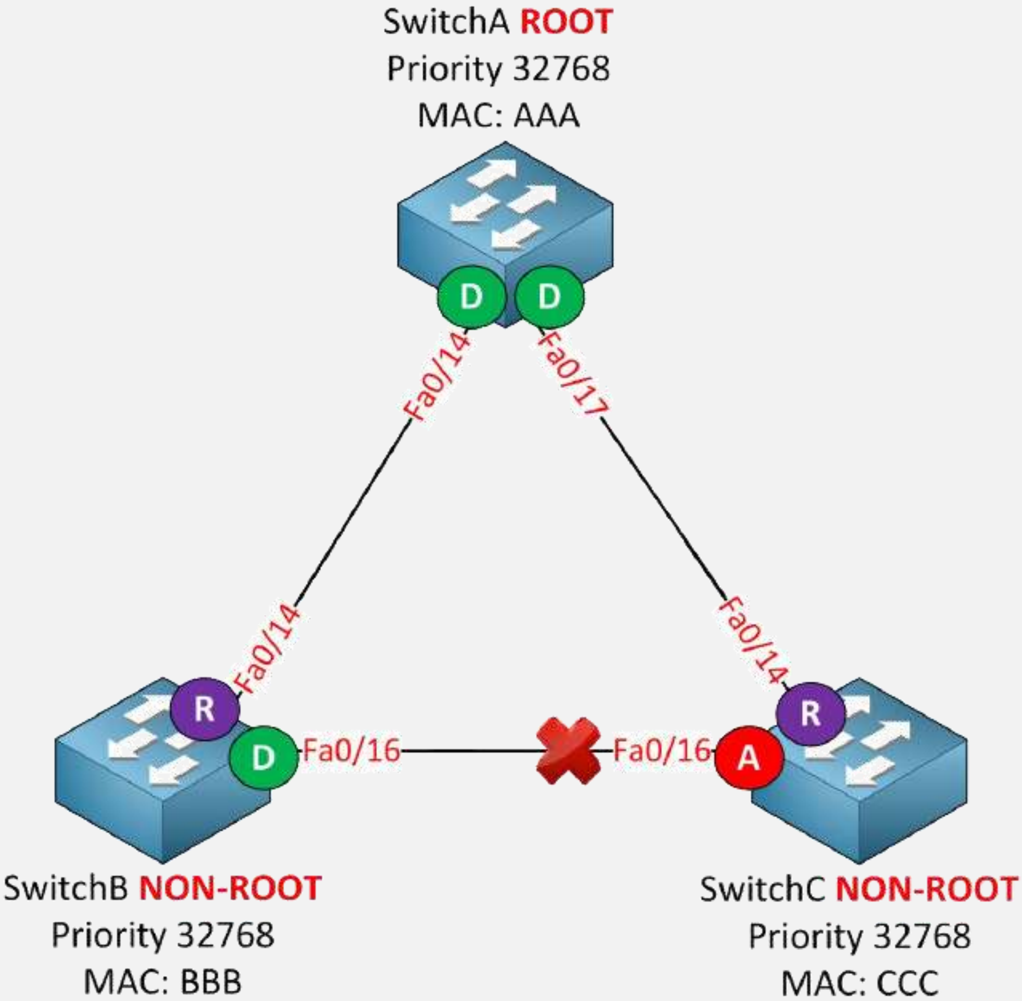


SwitchA is now the **root bridge** because it has the best bridge identifier. All the other switches are **called non-root**. **Interfaces that forward traffic are called designated ports in spanning-tree**. On a root bridge the interfaces are always in **forwarding** mode.



All the **non-root switches have to find the shortest path to the root bridge** .

The interface that **leads us to the root bridge** is **called the root port and is forwarding traffic.**



In order to break the loop we **have to block an interface between SwitchB and SwitchC.**

which one are we going to block? SwitchB and SwitchC will duke it out by comparing their bridge identifier. Keep in mind the bridge identifier consists of the priority and MAC address. The lowest bridge identifier is the best one, **SwitchB and SwitchC have the same priority but SwitchB has a lower MAC address. SwitchB will win this battle and as a result the fa0/16 of SwitchC will be blocked.**

A port that is blocking traffic is called an alternate port. The fa0/16 interface of SwitchB will become a designated port.

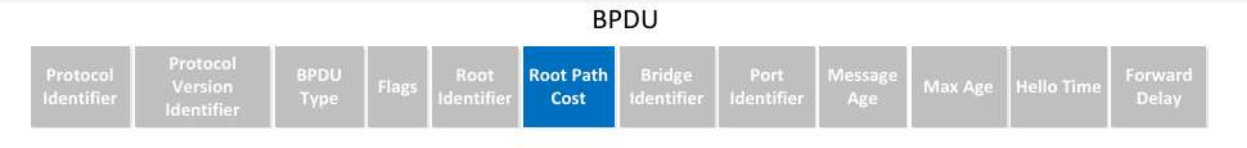
<u>Non-root bridges need to find the shortest path to the root bridge.</u>

We determine the shortest path or the fastest path based on

- **The lowest cost per interface.**
- **The lowest bridge ID.**
- **The lowest port priortiy (for down stream switch (the next switch)).**
- **The lowest port ID (for down stream switch (the next switch)).**

Spanning-tree **uses cost to determine the shortest path to the root bridge**. The **slower the interface, the higher the cost is**. **The path with the lowest cost will be used to reach the root bridge.**

Speed	STP cost	RSTP cost
10 Mbps (Ethernet)	100	2,000,000
100 Mbps (Fast Ethernet)	19	200,000
1 Gbps (Gigabit Ethernet)	4	20,000
10 Gbps (Gigabit Ethernet)	2	2000
100 Gbps	x	200
1 Tbps	x	20
10 Tbps	x	2



In the BPDU you can see a field **called root path cost**. **This is where each switch will insert the cost of its shortest path to the root bridge**. **Once the switches found out which switch is declared as root bridge they will look for the shortest path to get there.**

Types of Spanning tree

PROTOCOL	IEEE STANDARD	SWITCH	DESCRIPTION
Spanning Tree Protocol (STP)	IEEE 802.1D	stp	The original STP version
Rapid STP (RSTP)	IEEE 802.1w	rstp	An evolution of STP 802.1D that addresses the STP convergence time gap issue with enhanced BPDU exchange
Multiple STP (MSTP)	IEEE 802.1s	mstp	A format for mapping multiple VLANs into the same spanning tree to reduce processing on the switch
Per-VLAN Spanning Tree (PVST+)	Cisco protocol based on 802.1D	pvst	An 802.1D enhancement that provides a separate STP instance for each VLAN configured in the network
Rapid PVST+	Cisco protocol based on 802.1w	rapid-pvst	An 802.1w enhancement that provides a separate STP instance for each VLAN, enabling faster convergence times

Rapid STP (IEEE 802.1W)

Rapid spanning tree protocol (RSTP) is **a faster transition to a port-forwarding state.** **Unlike STP, which has five switchport states, RSTP has only three: discarding, learning, and forwarding.**

not supported on cisco devices.

RSTP Port States

Disabled - The result of an **administrative command** that will disable the port.

Discarding - When a **device is connected, the port will first enter the discarding state.**

Learning - The switch **will receive a BPDU,** will stop sending its own BPDUs.

Forwarding - The port is **forwarding traffic.**

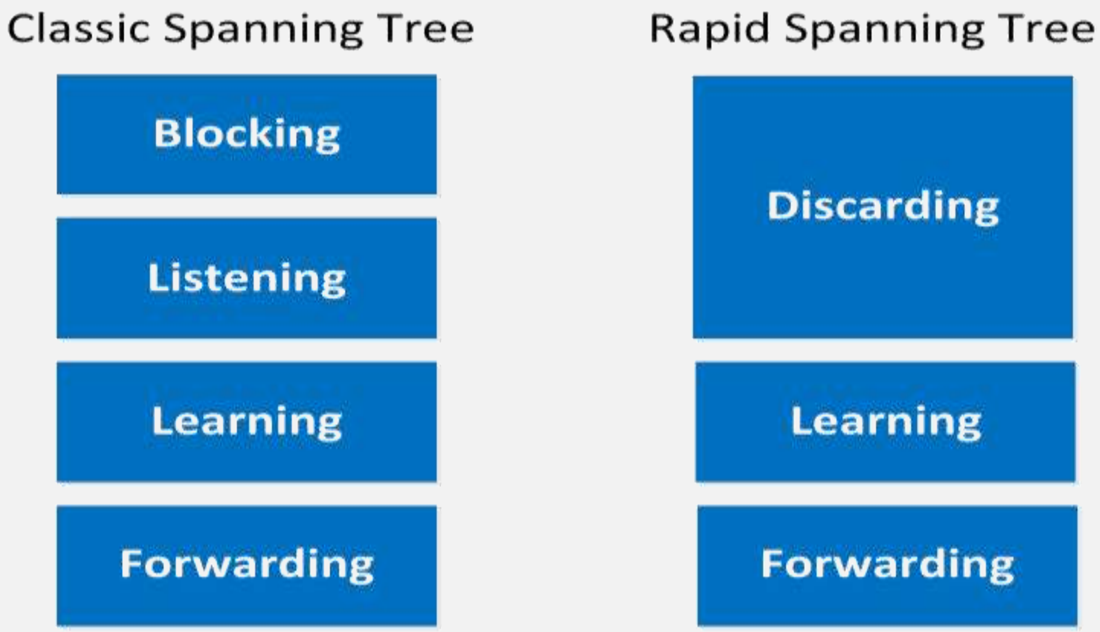
RSTP Port Roles

Root - **Ports on non-root switches with best cost path to root bridge**. These ports forward data to the root bridge.

Designated - Ports on **root switches**. **All ports on the root bridge will be designated.**

Alternate - **Receives BPDUs from another switch but remains in a blocked state.**

Backup - **Receives BPDUs from its own switch but remains in a blocked state**. **You are not likely to see this in a production environment unless hubs are used.**

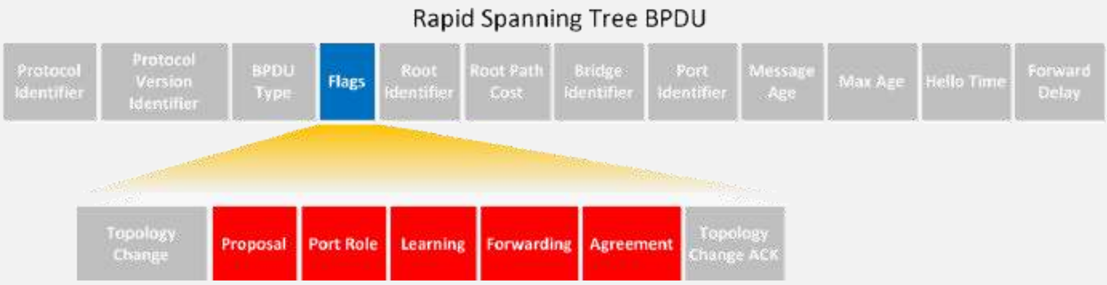


STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable

Note

The BPDU is different for rapid spanning tree. In the classic spanning tree the flags field only had two bits in use:

- **Topology change.**
- **Topology change acknowledgment.**



All bits of the **flag field are now used**. The role of the port that originates the BPDU will be added by **using the port role field**, it has the following options:

- **Unknown**
- **Alternate / Backup port.**
- **Root port.**
- **Designated port.**

Note:This new BPDU is called a **version 2 BPDU**. Switches running the **old version of spanning tree** will **drop this new BPDU version**.

Difference between RSTP and STP work

BPDUs are now sent **every hello time**. **Only the root bridge generated BPDUs in the classic spanning tree**. **Rapid spanning tree works differently** ***all switches generate BPDUs every two seconds (hello time).***

Rapid spanning tree send ***BPDU as a keepalive update***. **If a switch misses three BPDUs from a neighbor switch it will assume connectivity to this switch has been lost and it will remove all MAC addresses immediately.**

In STP Every switch send hello message every 2 sec as a keepalive and the max age = 20 sec and to remove the PCs mac address this will take the time for lisenting and learning = 30 sec.

Transition speed (convergence time) is the most important feature of rapid spanning tree. **The classic spanning tree** had to walk through **the listening and learning state before it would move an interface to the forwarding state**, this **took 30 seconds** with the default timers.

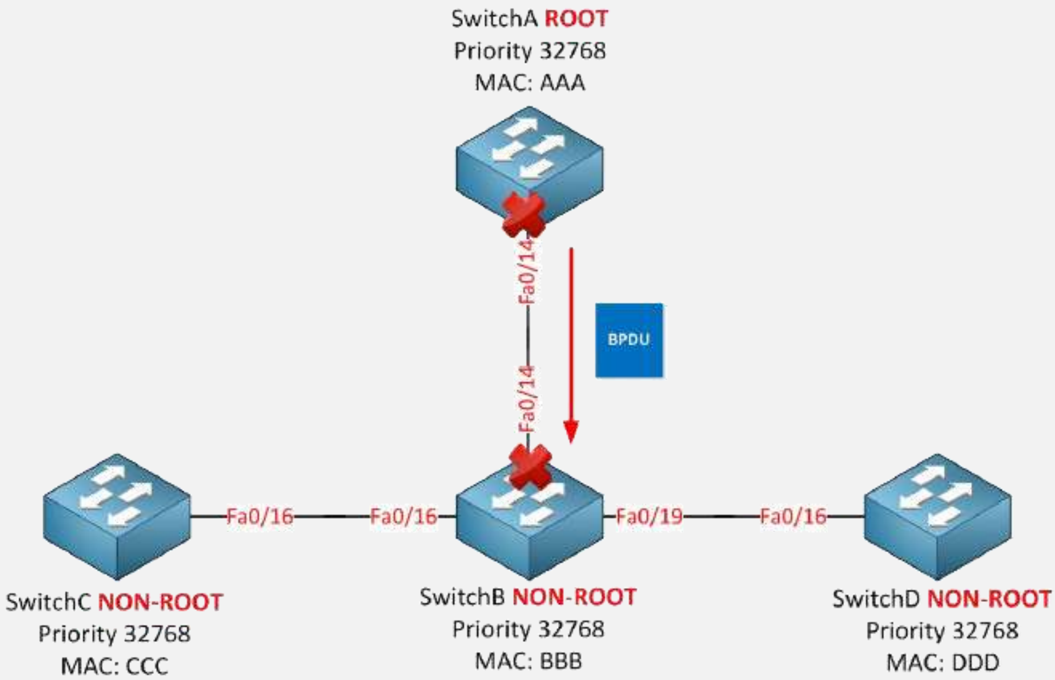
Note: **Rapid spanning doesn't use timers** to decide whether an interface can move to the forwarding state or not. **It will use a negotiation mechanism for this**.

- **If we enable portfast while running the classic spanning tree** it **will skip the listening and learning state and put the interface in forwarding state immediately.** Besides **moving the interface to the forwarding state it will also not generate topology changes notification when the interface goes up or down**. **We still use portfast for rapid spanning tree** but **it's now referred to as an edge port.**
- **Rapid spanning tree** can **only put interfaces in the forwarding state really fast on edge ports (portfast) OR point-to-point interfaces**. **there are only two link types:**
  - **Point-to-point (full duplex)**
  - **Shared (half duplex)**

Normally we are using switches and all our interfaces are configured **as full duplex, rapid spanning tree sees these interfaces as point-to-point**. If we introduce a **hub to our network we'll have half duplex which is seen as a shared interface to rapid spanning-tree.**

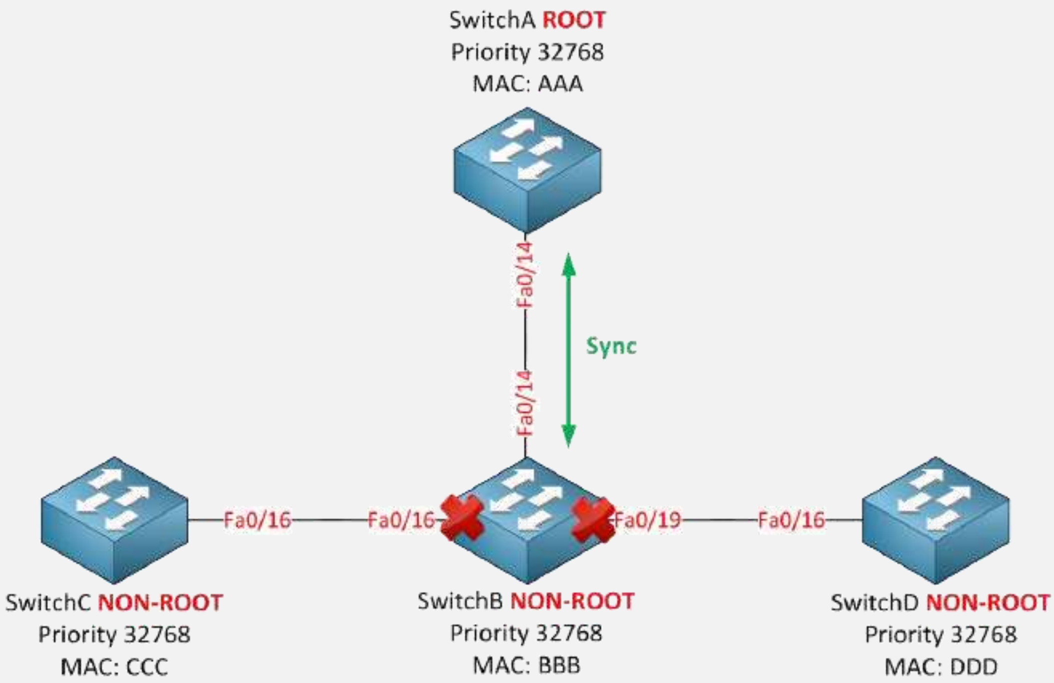
sync mechanism in RSTP

As soon as the link between SwitchA and SwitchB comes up their interfaces will be in blocking mode. **SwitchB will receive a BPDU from SwitchA and now a negotiation will take place called sync.**



After SwitchB received the BPDU from the root **bridge it immediately blocks all its non- edge designated ports.**

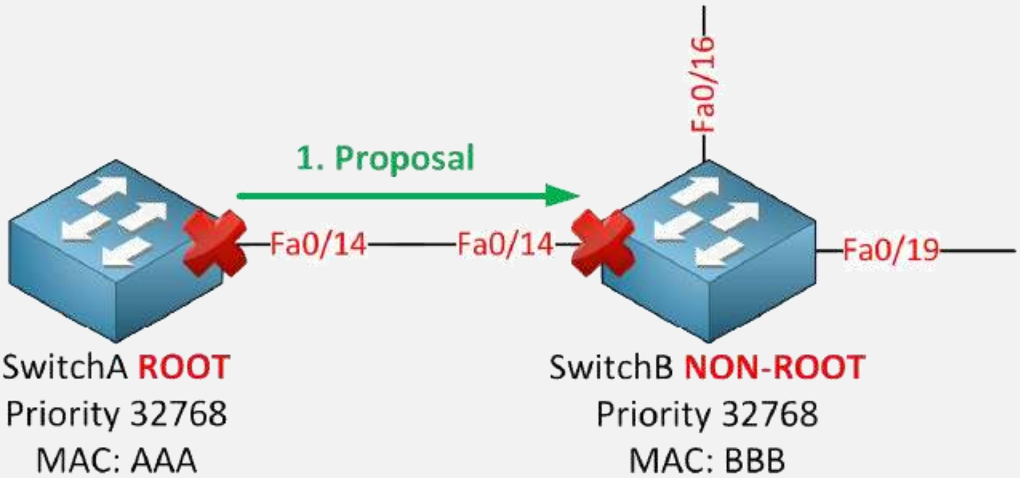
Non-edge ports are the interfaces that connect to other switches while edge ports are the interfaces that have portfast configured. As soon as SwitchB blocks its non-edge ports the link between SwitchA and SwitchB will go into forwarding state.



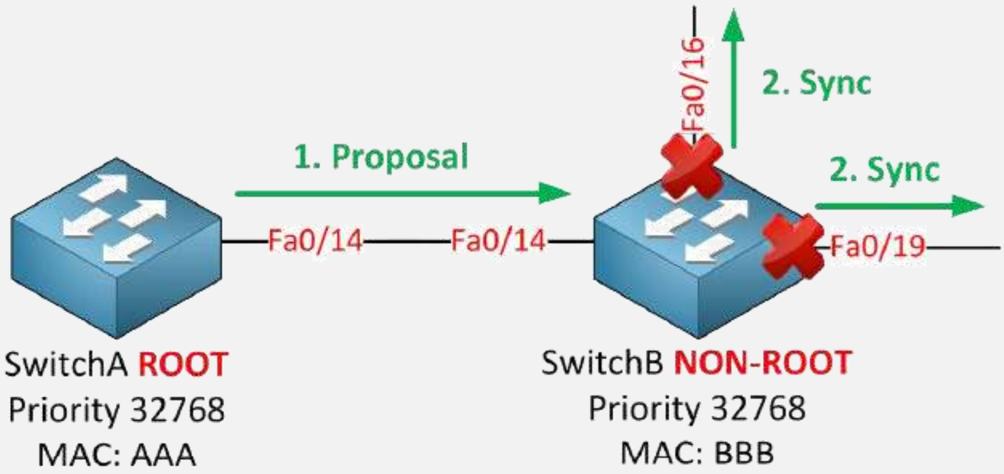
SwitchB will also perform a sync operation with both SwitchC and SwitchD so they can quickly move to the forwarding state.

then **rapid spanning tree uses this sync mechanism instead of the "timer-based" mechanism** that the classic spanning tree uses (listening ➔learning➔ forwarding).

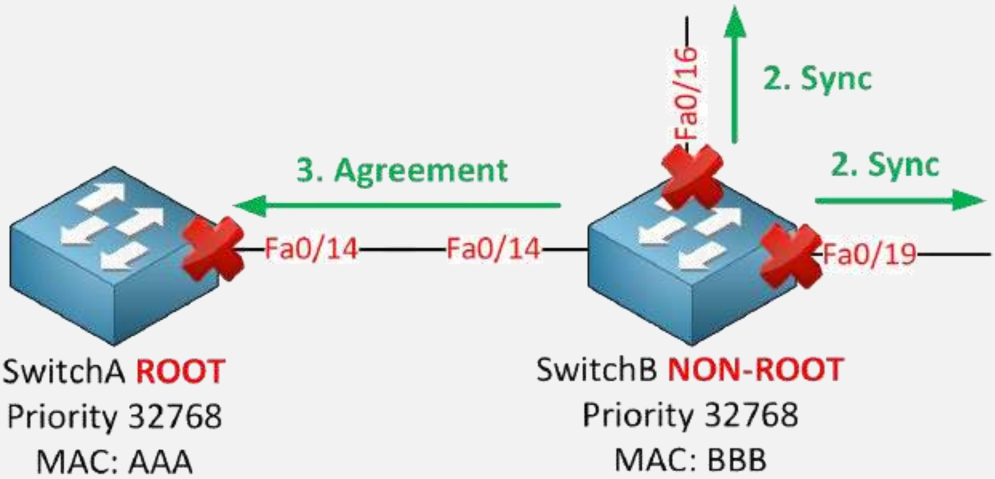
<u>sync mechanism in RSTP</u>



At first the interfaces will be blocked until they receive a BPDU from each other. At this moment SwitchB will figure out that SwitchA is the root bridge because it has the best BPDU information. The sync mechanism will start because SwitchA will set **the proposal bit in the flag field of the BPDU.**

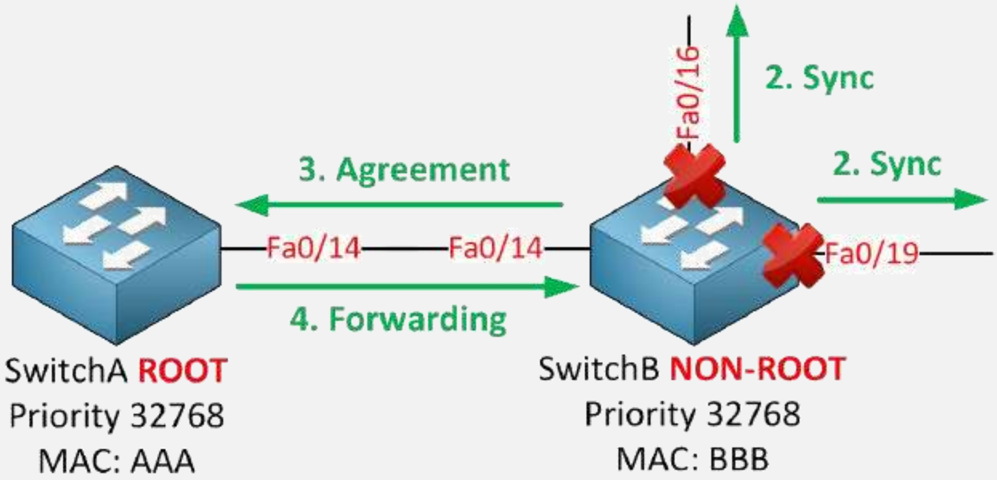


SwitchB receives the proposal from SwitchA and realizes it has to do something. It will block all its non-edge interfaces and will start the synchronization towards SwitchC and SwitchD.



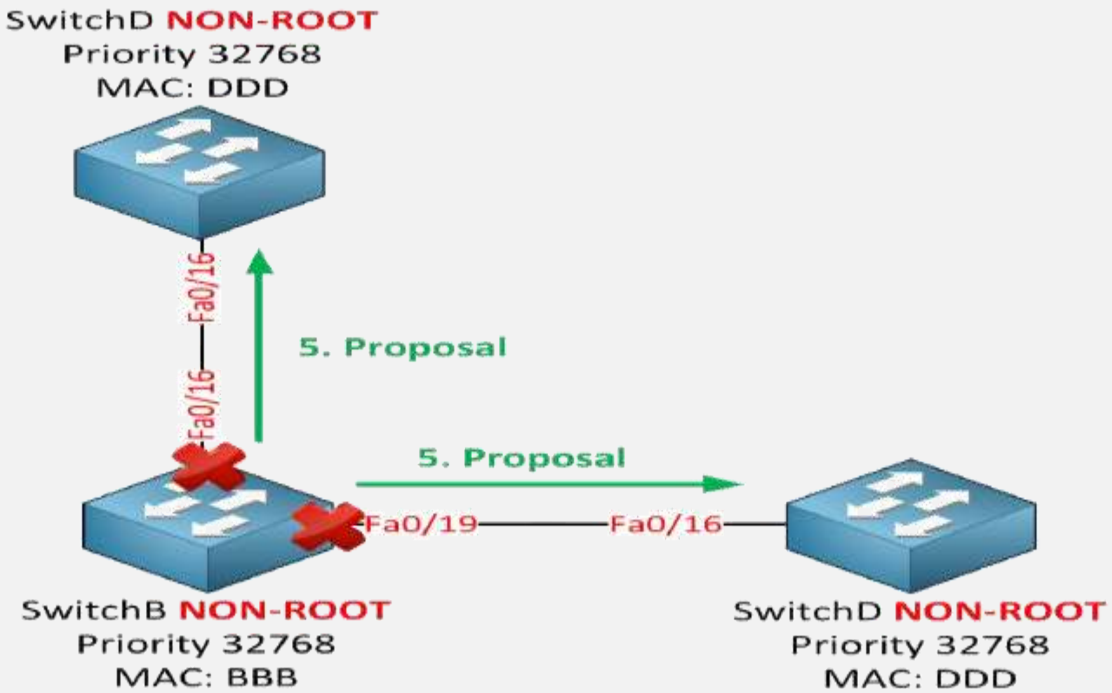
Once SwitchB has its interfaces in sync mode it will let **SwitchA know about this by sending an agreement.**

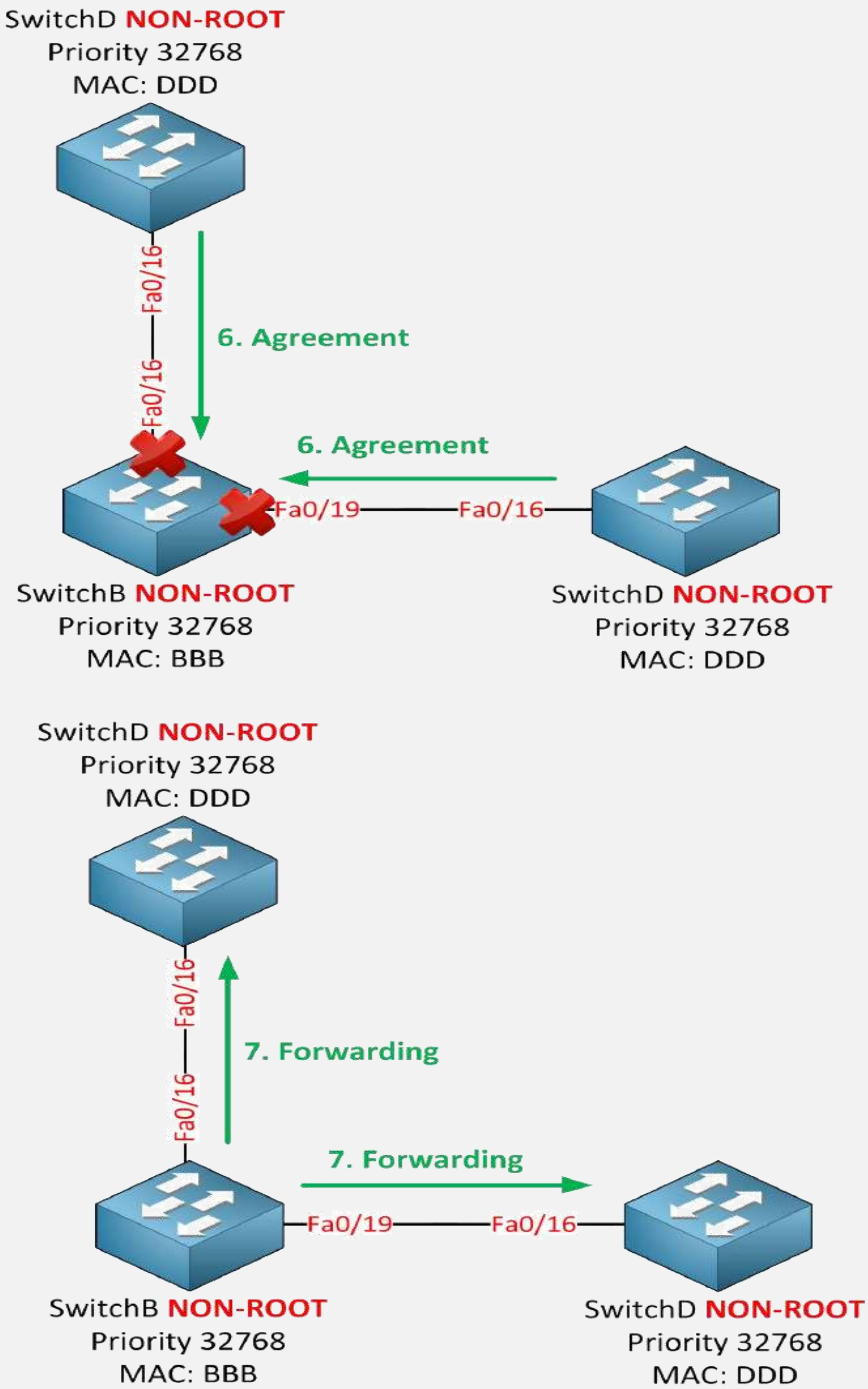
This agreement is a copy of the proposal BPDU where the proposal bit has been switched off and the agreement bit is switched on. The fa0/14 interface on SwitchB will now go into forwarding mode.



Once SwitchA receives the agreement from SwitchB it will put its fa0/14 interface in forwarding mode immediately.

The exact same sync mechanism will take place now on these interfaces. SwitchB will send a proposal on its fa0/16 and fa0/19 interfaces towards SwitchC and SwitchD.




SwitchD NON-ROOT
Priority 32768
MAC: DDD
Fa0/16
Fa0/16
6. Agreement
6. Agreement
Fa0/19
Fa0/16
SwitchB NON-ROOT
Priority 32768
MAC: BBB
SwitchD NON-ROOT
Priority 32768
MAC: DDD
SwitchD NON-ROOT
Priority 32768
MAC: DDD
Fa0/16
Fa0/16
7. Forwarding
7. Forwarding
Fa0/19
Fa0/16
SwitchB NON-ROOT
Priority 32768
MAC: BBB
SwitchD NON-ROOT
Priority 32768
MAC: DDD

Notes:

- When you configure the classic spanning tree you have to enable UplinkFast yourself. Rapid spanning tree uses UpLinkFast by default.
- rapid spanning tree and classic spanning tree are compatible.
- Topology change mechanism.

The difference is that the classic spanning tree needed multicast frames to update the MAC address tables of all switches. We don't need this anymore because the topology change mechanism for rapid spanning tree is different.

what's different about the topology change mechanism?

With the ***classic spanning tree*** a link failure would trigger a **topology change**. ***Using rapid spanning tree*** a **link failure is not considered as a topology change**. Only ***non-edge interfaces*** (leading to other switches) that move to the forwarding state are considered as a topology change. Once a switch detects a topology change this will happen:

- It will start a **topology change while timer** with a value that is twice the hello time. This will be done for all non-edge ports.
- It **will flush the MAC addresses** that are learned on these ports.
- As long as the topology change while timer is active it will set ***a topology change bit on BPDUs*** that are sent out these ports. (BPDUs will also be sent out of its root port).

When a neighbor switch receives this BPDU with a topology change bit set this will happen:

- It **will clear all its MAC addresses on all interfaces except the one where it received the BPDU with the topology change on.**

Per-VLAN Spanning Tree Plus (PVST+)

allows a separate instance of STP to run for each Virtual LAN (VLAN) in the network. This means that if you have ten VLANs, you can have ten different spanning trees (A Root-Bridge for each VLAN).

PVST+ supports dot1q but it is still slow like classic STP. In both protocols, the max age timer is 20 seconds, and the listening and learning states are 15 seconds each, so it can take up to 50 seconds to respond to changes in the network.

Per-VLAN Spanning Tree Plus (PVST+)
- Cisco's upgrade to 802.1D
- Each VLAN has its own STP instance.
- Can load balance by blocking different ports in each VLAN.

Rapid Per-VLAN Spanning Tree Plus (Rapid PVST+)
- Cisco's upgrade to 802.1w
- Each VLAN has its own STP instance.
- Can load balance by blocking different ports in each VLAN.

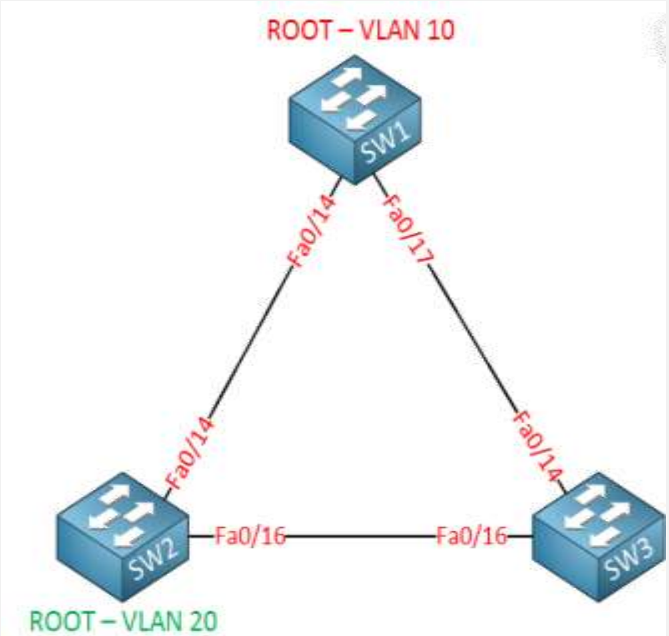
PVST+= ieee (this not mean stander IEEE)

The priority for the switch is equal 32768 by default + system-id extended (vlan number).

we can create a different root bridge for each VLAN. SW1 could be the root bridge for VLAN 10, and SW2 could be the root bridge for VLAN 20.

We can also configure another switch to become the "secondary" root bridge.

```
SW1(config)#spanning-tree vlan 10 root ?
  primary    Configure this switch as primary root for this spanning tree
  secondary  Configure switch as secondary root
```



Rapid PVST+

- **An enhancement of PVST+, Rapid PVST+ is based on the Rapid Spanning Tree Protocol (RSTP), which is a faster version of STP.**
- **It provides faster convergence than PVST+ by quickly reconfiguring the spanning tree in case of topology changes.**
- **Rapid PVST+ maintains a separate spanning tree for each VLAN, just like PVST+.**

The configuration:

1. **Enable Rapid PVST+**:

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

2. **Configure the Root Bridge**:

```
switch(config)# spanning-tree vlan <VLAN_ID> root primary
```

3. **Configure a Secondary Root Bridge**:

```
switch(config)# spanning-tree vlan <VLAN_ID> root secondary
```

4. **Set Port Priority**:

```
switch(config-if)# spanning-tree vlan <VLAN_ID> port-priority <priority_value>
```

MST (Multiple Spanning Tree)(IEEE 802.1s)

Multiple Spanning Tree Protocol is defined in IEEE 802.1s standard. MSTP is based on RSTP, which means it inherits all the features and improvements of RSTP, such as fast convergence, port roles, port states, edge ports, link types, etc. But, MSTP added some new ones, such as regions, instances, mapping, and digest.

MSTP allows for creating multiple spanning trees within a single network. Each spanning tree is called an instance, and each instance can have its own root bridge and forwarding topology. Each instance can also be associated with one or more VLANs, which means that different VLANs can follow different paths on the same physical network. This can be useful for larger networks that require more flexibility and redundancy.

The main purpose of MSTP is to optimize network performance and reliability by reducing the number of spanning tree instances and allowing different VLANs to follow different paths. MSTP also simplifies network configuration and management by using a common set of parameters for all instances within a region.

Define MST region

An MST region is defined as the group of switches that run MSTP and have the same configuration parameters. These parameters include:

Region name: A name that identifies the region.

Revision number: A number that indicates the version of the configuration.

VLAN-to-instance mapping: A table that assigns each VLAN to an instance.

Within the MST region, we will have one instance of spanning tree ***that will create a loop-free topology within the region***. When you configure MST, there is always one default instance used to calculate the topology within the region. We call this the IST (Internal Spanning Tree). By default, Cisco will use instance 0 to run the IST. The IST runs rapid spanning tree. (instance 0 used for the rest of vlans that didn't enter in instance)

Note:

Inside the same region RSTP work but between different region stp work .

In the region there is a root for this region. and there is a root for the all regions (network) (called regional root sw).

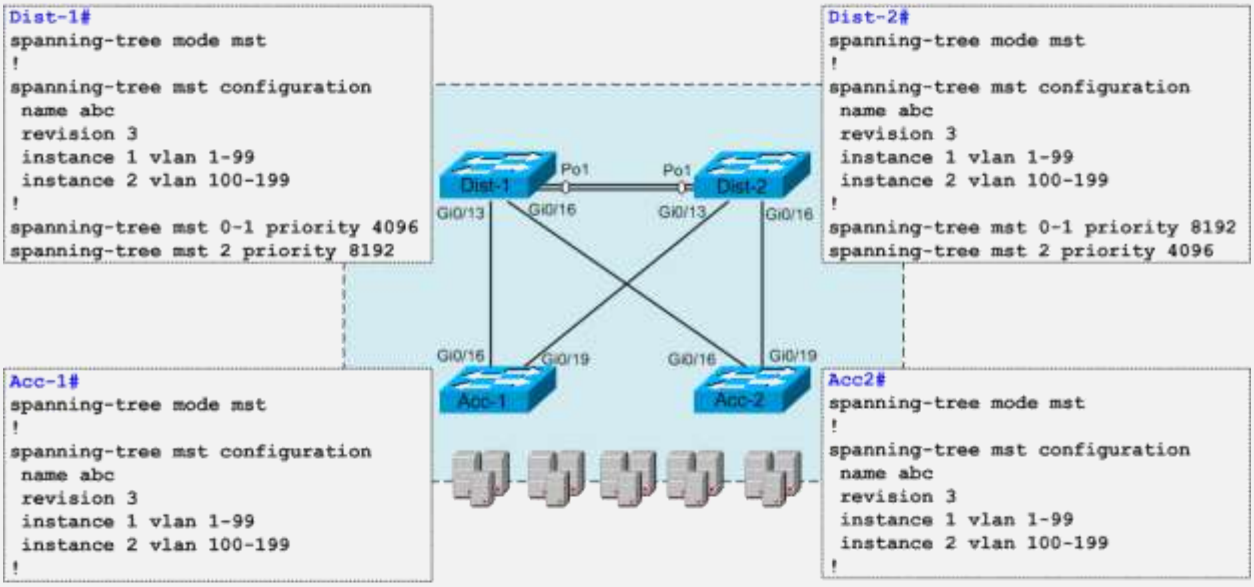
The same instance means has it's own calculation.

How to know the swithces in the same region?

1-swicthes have the same name of region

2-the same revision number

3-the same mapping between vlan databaxe and instance number.



The instance communicate with each other using STP .

There is root bridge per region.