Spanning Tree Protocol – Detailed Operation and Configuration

Spanning Tree Port States

Spanning Tree Protocol (STP) defines four port states, two of which are stable and two that are transitional. The two stable states are **Blocking** and **Forwarding**. Root ports and designated ports remain stable in the Forwarding state, while non-designated ports remain stable in the Blocking state – as long as the network topology does not change. The two transitional states are **Listening** and **Learning**, which a port passes through when it is activated or when a blocking port must transition to forwarding due to a topology change.

A fifth state, **Disabled**, refers to an interface that is administratively shut down. It plays no active role in STP.

STP Port State	Stable/Transitional
Blocking	Stable
Listening	Transitional
Learning	Transitional
Forwarding	Stable

(Disabled)

Blocking State

STP Port State	Stable/Transitional
Blocking	Stable

- Non-designated ports are in the Blocking state.
- Interfaces in this state are effectively disabled to prevent loops.
- They **do not send or receive regular network traffic**; any regular frames arriving are dropped.
- They **do receive and process STP BPDUs** to stay aware of the spanning tree topology.
- They **do not forward STP BPDUs**.
- They **do not learn MAC addresses** from arriving frames.

Listening State

STP Port State	Stable/Transitional
Listening	Transitional

- After the Blocking state, interfaces with Desginated or Root role enter the Listening state.
- Only Designated or Root ports enter the Listening state when they begin transitioning toward forwarding. Non-designated ports remain blocked.
- The Listening state lasts **15 seconds** by default (determined by the *Forward delay* timer).
- An interface in Listening **only sends/receives STP BPDUs** – regular traffic is discarded.
- An interface in the listening state does NOT send/receive regular traffic that arrives on the interface.
- **No MAC address learning** occurs.

Learning State

STP Port State	Stable/Transitional
Learning	Transitional

- After Listening, Designated or Root ports enter the Learning state.
- This state also lasts **15 seconds** by default (same Forward delay timer).
- Like Listening, the interface **only sends/receives STP BPDUs** and does not forward regular traffic.

- **However**, the interface **does learn MAC addresses** from regular traffic that arrives, building up the MAC address table in preparation for forwarding.

Forwarding State

STP Port State	Stable/Transitional
Forwarding	Stable

- Root and Designated ports remain in Forwarding when stable.
- A port in Forwarding operates normally:
  - Sends and receives BPDUs.
  - Sends and receives normal traffic.
  - Learns MAC addresses and populates the MAC address table.

Summary of STP Port States

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

State	Behaviour
Blocking	No regular traffic; receives BPDUs; no MAC learning; stable state for non-designated ports.
Listening	Transitional (15 s); only BPDUs; no MAC learning; for designated/root ports.
Learning	Transitional (15 s); only BPDUs; learns MAC addresses; for designated/root ports.
Forwarding	Stable; full operation (traffic, BPDUs, MAC learning).
Disabled	Administratively shut down – no STP participation.

Spanning Tree Timers

Three key timers control the operation of STP.

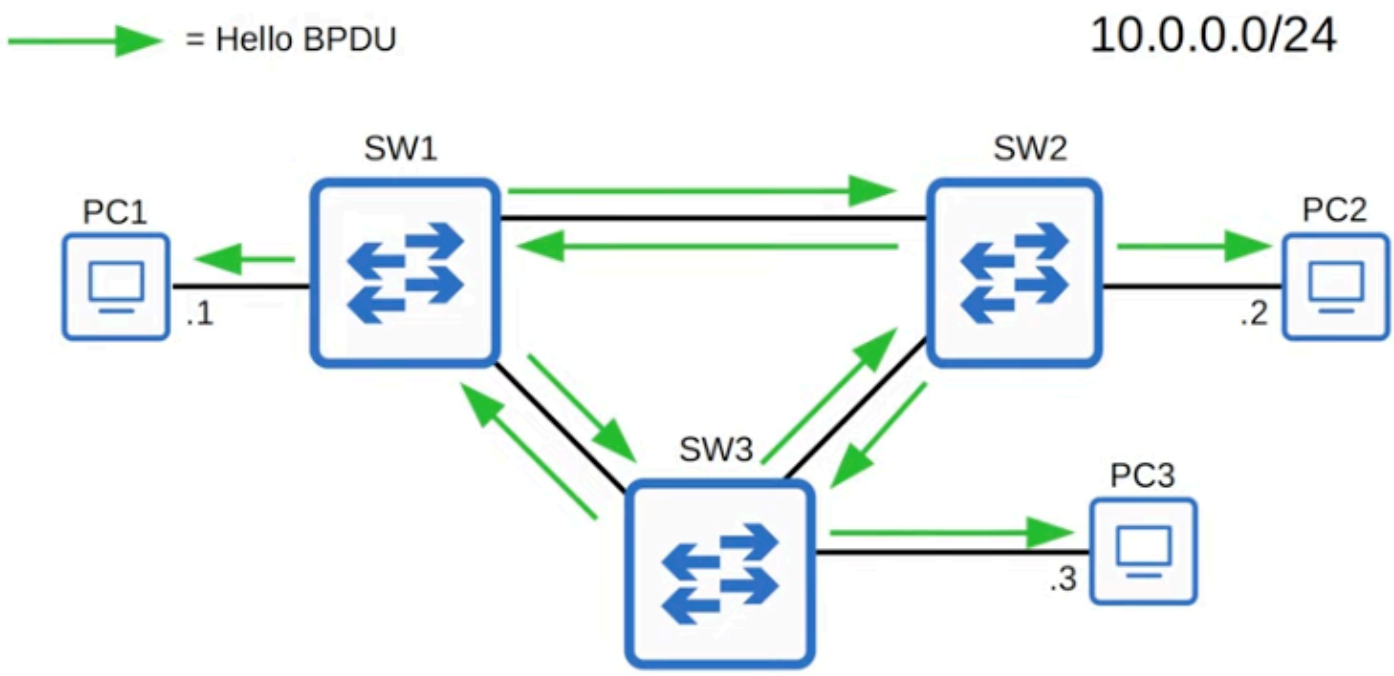
STP Timer	Purpose	Duration
Hello	How often the root bridge sends hello BPDUs	2sec
Forward delay	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
Max Age	How long an interface will wait after ceasing to receive Hello BDPUs to change the STP topology.	20sec (10* hello)

Hello Timer

1. Determines how often the **root bridge** sends hello BPDUs.
2. Default: **2 seconds**.
3. Other switches do not originate BPDUs; they only forward them on their **Designated ports** (not on root ports or non-designated ports).

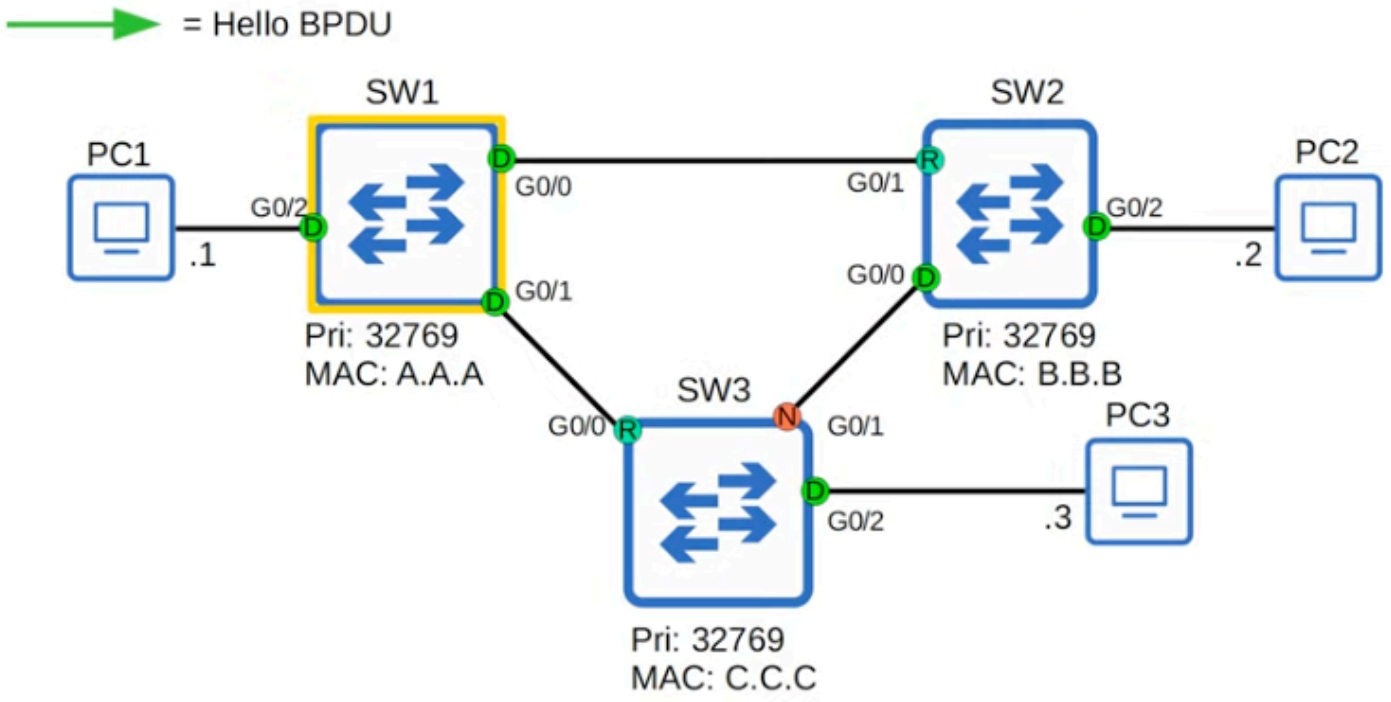
Let's see how that works:

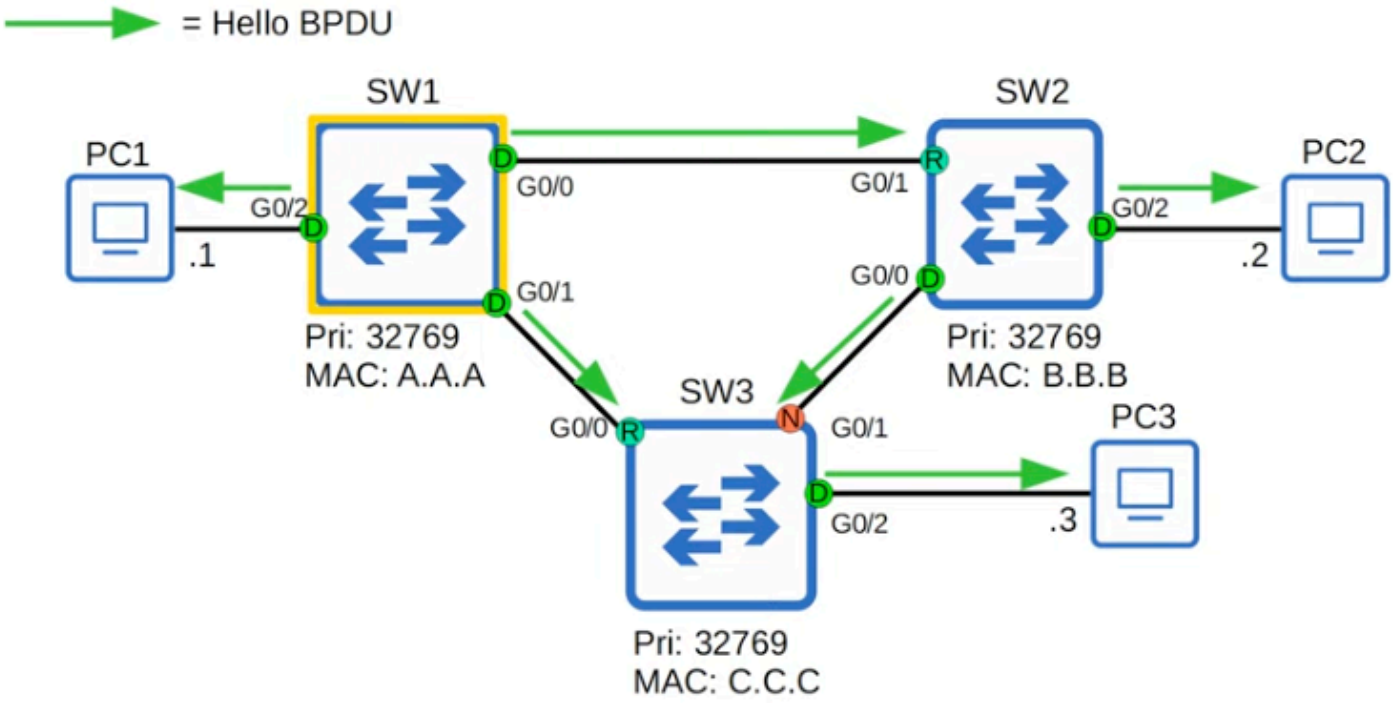
1. **Initial State: All Switches Think They're the Root**



- **What happens:** When all switches power on at the same time, there is no "boss" yet.
- **Action:** Every switch assumes it is the root bridge and begins sending BPDUs out of *all* of its interfaces.
- **Result:** This is a temporary, chaotic state before the network "converges" (stabilizes).

2. Converged State: Only One Sends, Others Forward





- **What happens:** After STP calculations finish, the network stabilizes.
- **Action:** Only the **Root Bridge** originates and sends BPDUs.
- **Action by other switches:** Non-root switches do **not** create their own BPDUs. Instead, they **relay or forward** the BPDUs they receive from the root bridge.
- **Key update:** When they forward these BPDUs, they update the information in the BPDU (like the path cost, their own Bridge ID, and Port ID).

Forward Delay Timer

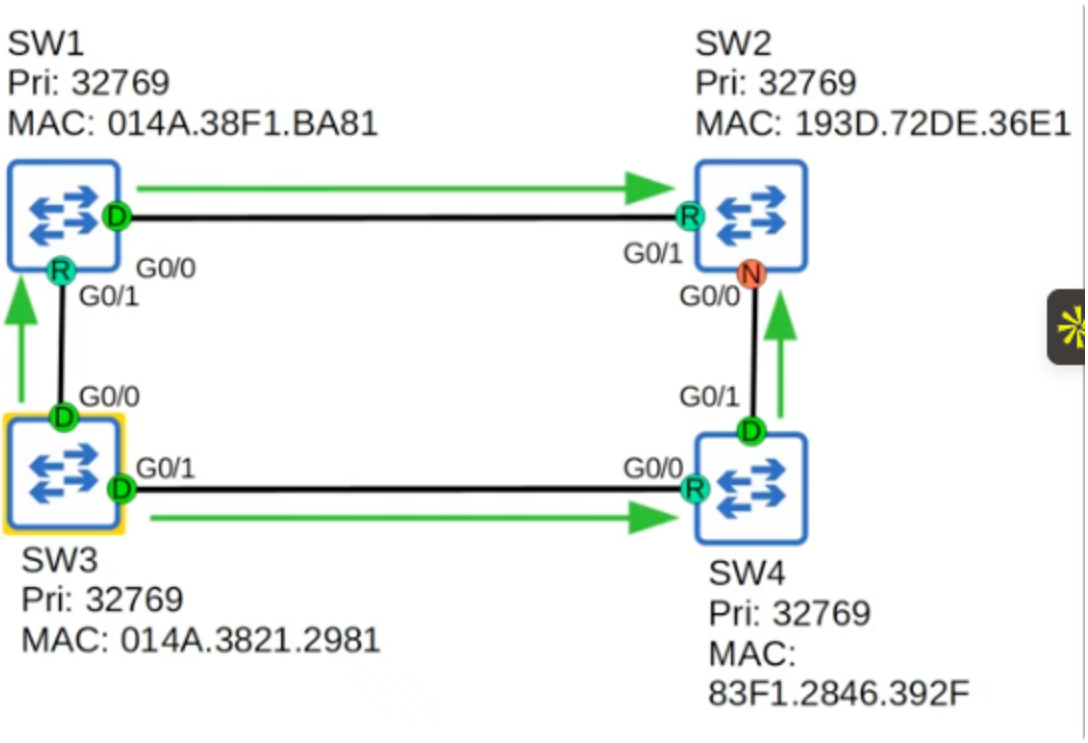
- Determines the length of both the **Listening** and **Learning** transitional states.
- Default: **15 seconds** per state, so a total of 30 seconds to move through both states before forwarding begins.

Max Age Timer

- Indicates how long an interface will wait to change the STP topology after it stops receiving BPDUs.
- Default: **20 seconds**.
- When a BPDU is received, the timer resets to 20. If it counts down to 0 without receiving another BPDU, the switch reevaluates its STP choices (root bridge, port roles).
- A blocking port that becomes designated or root will then transition: Blocking → Listening (15 s) → Learning (15 s) → Forwarding. Thus it can take up to **50 seconds** for a blocking port to fully transition to forwarding.
- The timer exists to prevent accidental loops.

The timers of the **root bridge** determine the timers used by all other switches in the network, regardless of local configuration. They are to make sure that loops aren't accidentally created by an interface moving to forwarding state too soon.

1. The Rule: BPDUs are Only Sent out of Designated Ports



- **Concept:** Within a single collision domain (a cable connecting two switches), only one interface is the Designated Port (DP). That DP is responsible for forwarding BPDUs onto that link.
- **Other Roles:**
  - **Root Ports (RP):** They are the best path *toward* the root. They are "ears" that listen for BPDUs arriving on the link.
  - **Non-Designated Ports (NDP):** They are blocking. They are also "ears" that listen for BPDUs to stay aware of the topology.
- **The Key Point:** All ports (RP, DP, NDP) expect to **receive** BPDUs, but only the DP is allowed to **send/forward** them out onto the segment.

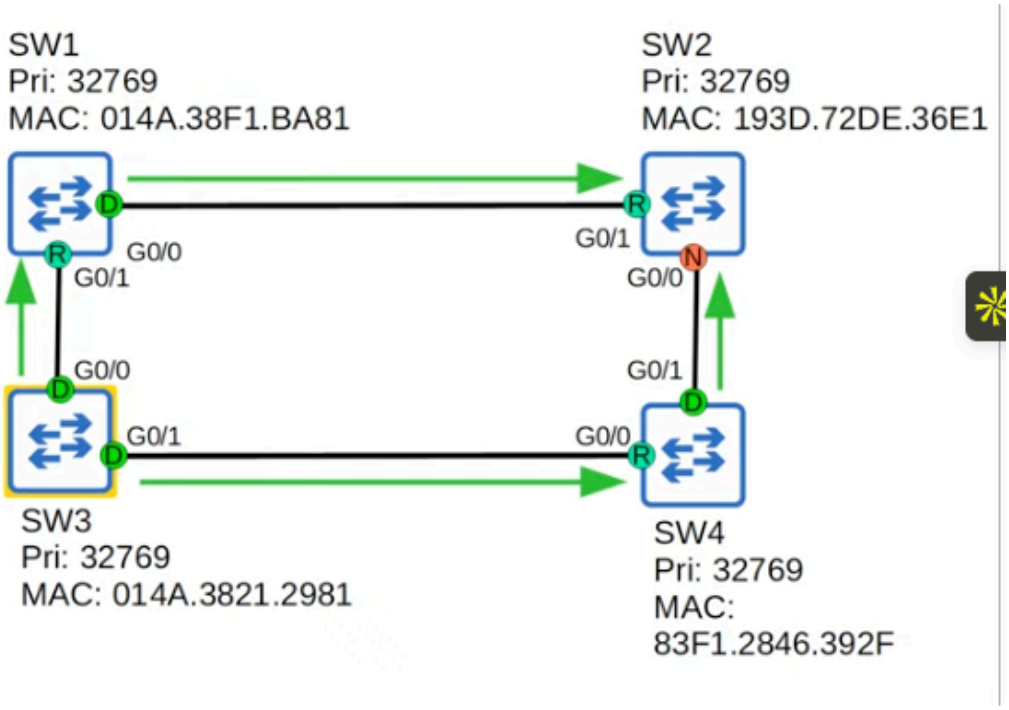
2. Normal Operation (The Max Age Timer Resets)

SW2 Max Age Timer:

20

19

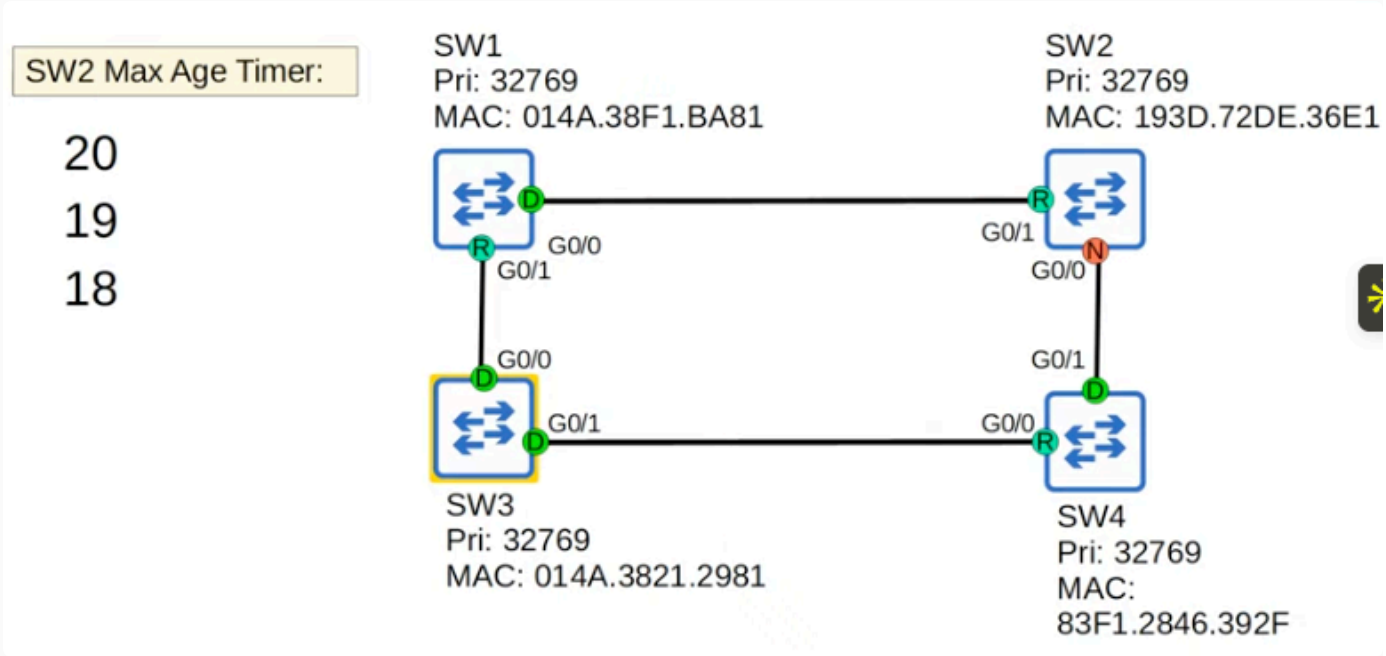
18



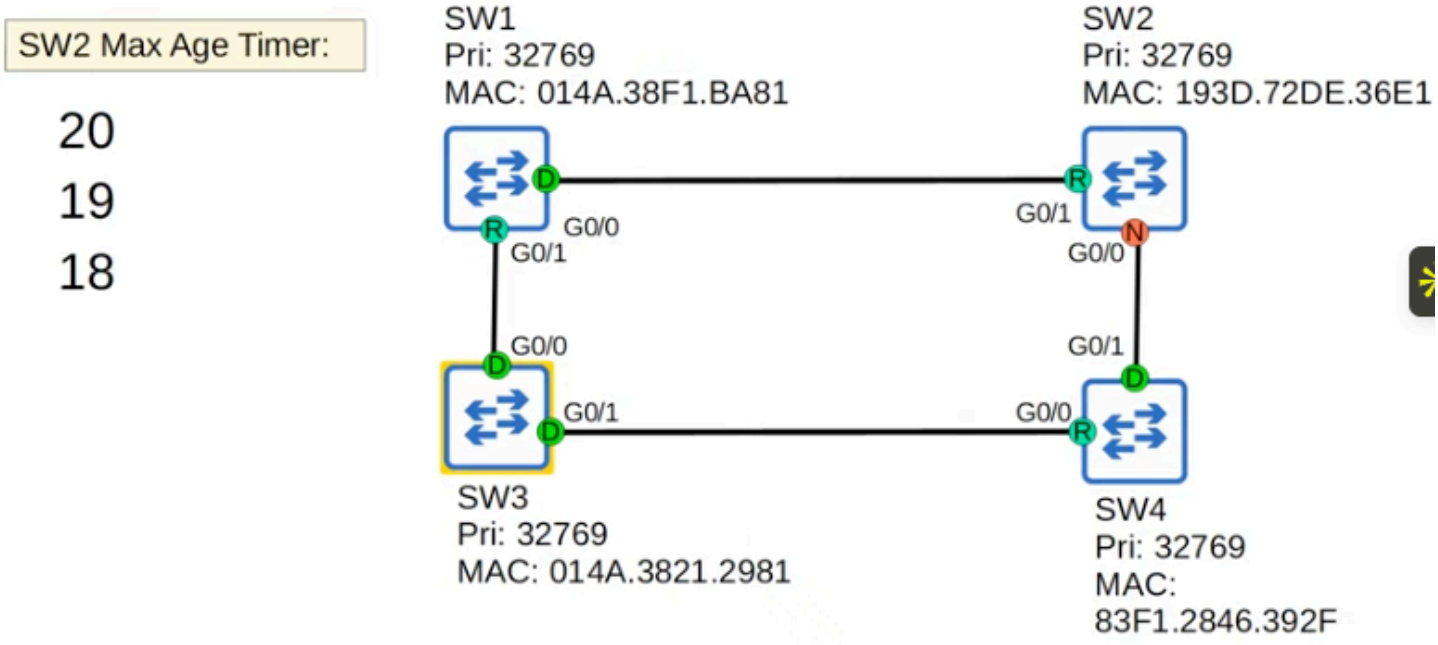
- **Setup:** SW3 is the root bridge. SW1 and SW4 are downstream switches with Designated Ports. SW2 has a Root Port (G0/1) connecting to SW1.

- **The Process:**

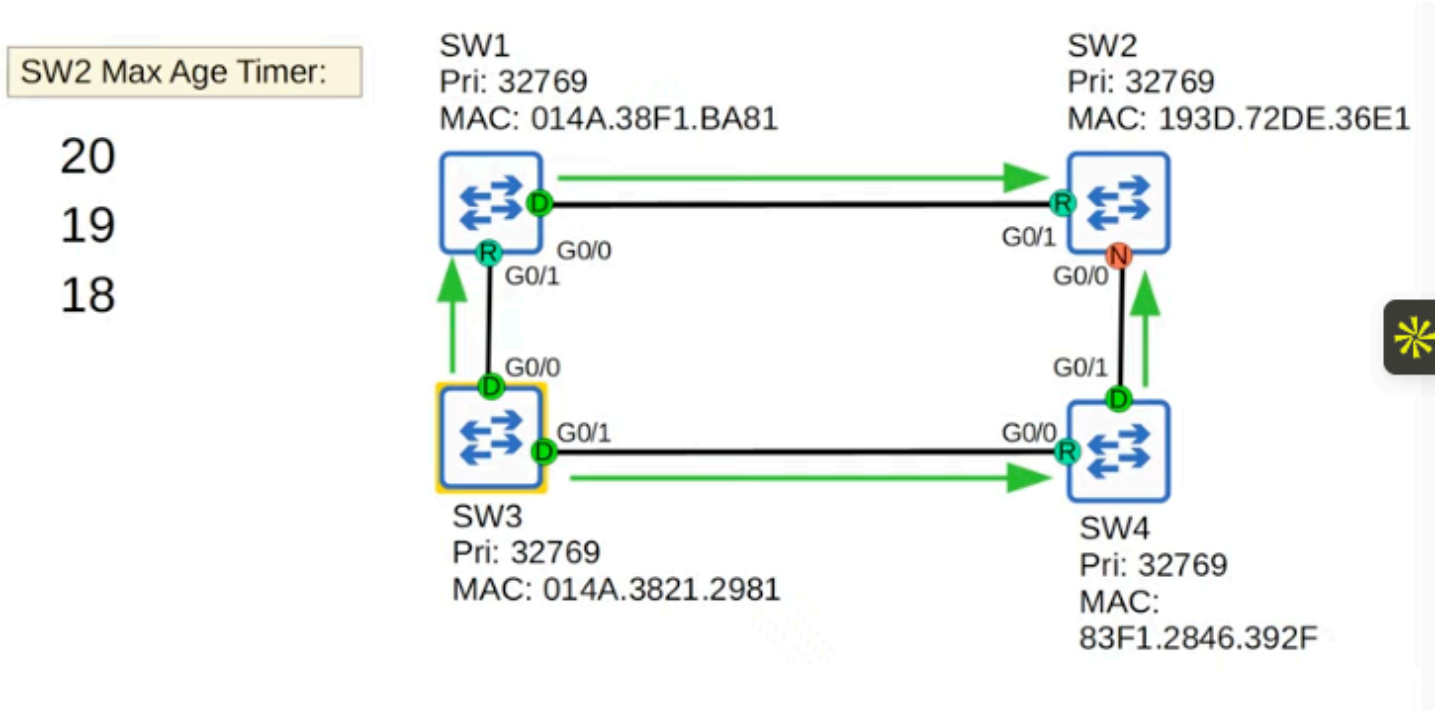
  1. Root bridge (SW3) sends a BPDU.
  2. SW1 receives it and forwards it out of its Designated Port (G0/0) towards SW2.
  3. SW2's G0/1 interface receives that BPDU.
  4. **Action:** This resets the **Max Age timer** on G0/1 back to the default of **20 seconds**.
  5. The timer then starts counting down: 20... 19... 18...



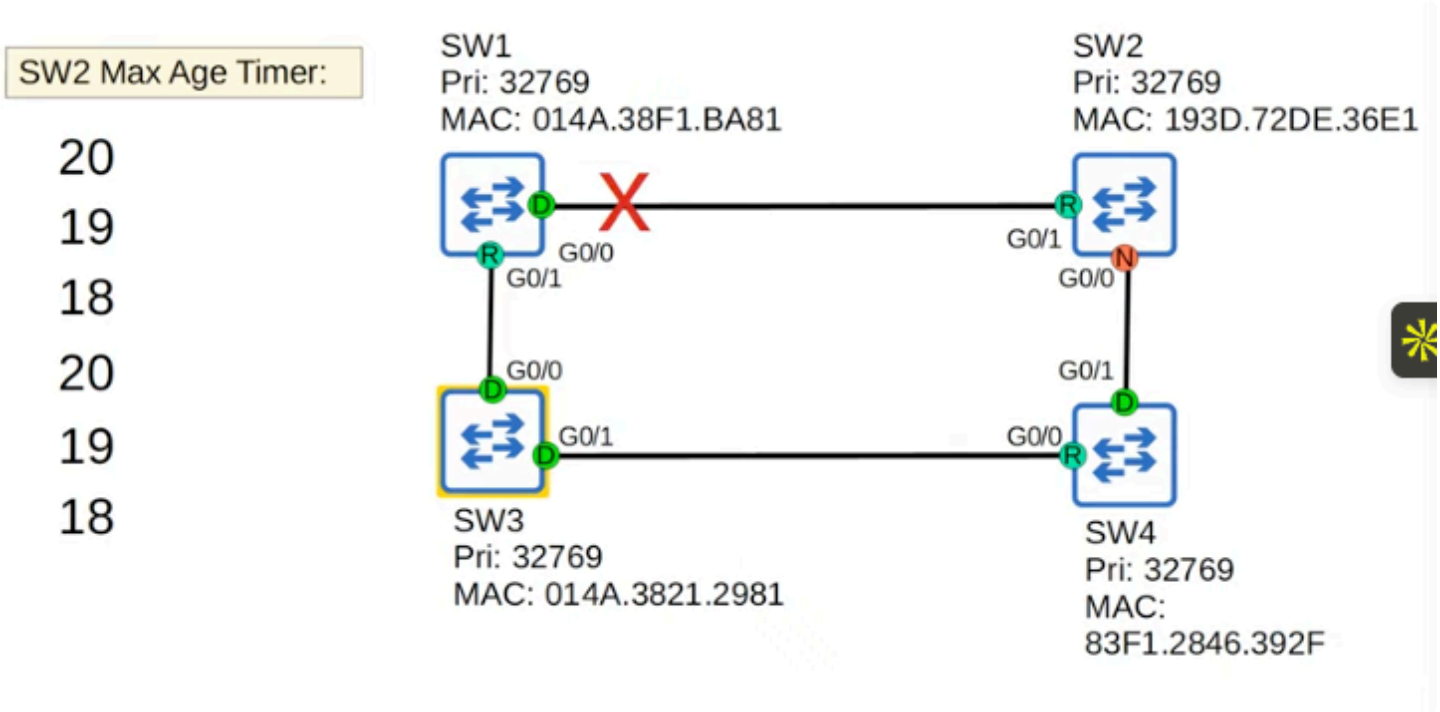
- 6. Before it reaches zero, the next BPDU is sent by the root (2 seconds later, thanks to the Hello timer), relayed by SW1, and received by SW2.



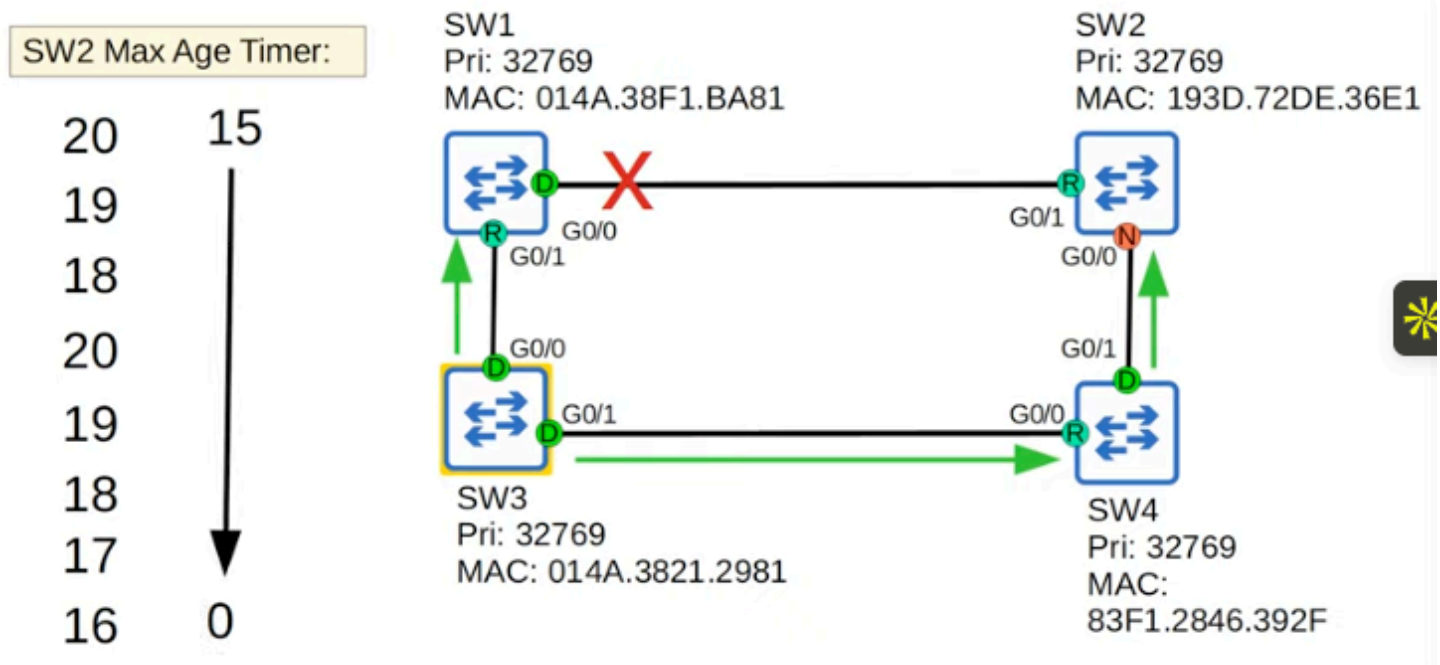
7. **Result:** The timer is reset to 20 again. This is a healthy, stable network.



3. The Failure Scenario (Max Age Counts Down to Zero)



- **The Event:** A failure occurs on the link between SW1 and SW2.
- **The Consequence:** The root bridge (SW3) continues sending BPDUs, and other switches (like SW4) forward them. But SW1's G0/0 is down, so **SW2's G0/1 interface stops receiving BPDUs.**



- **The Countdown:**
    1. SW2's Max Age timer is not reset (no BPDU arrives).
    2. It continues to count down: 17... 16... 15... and eventually reaches **0**.

4. The Reaction (When Max Age Hits Zero)

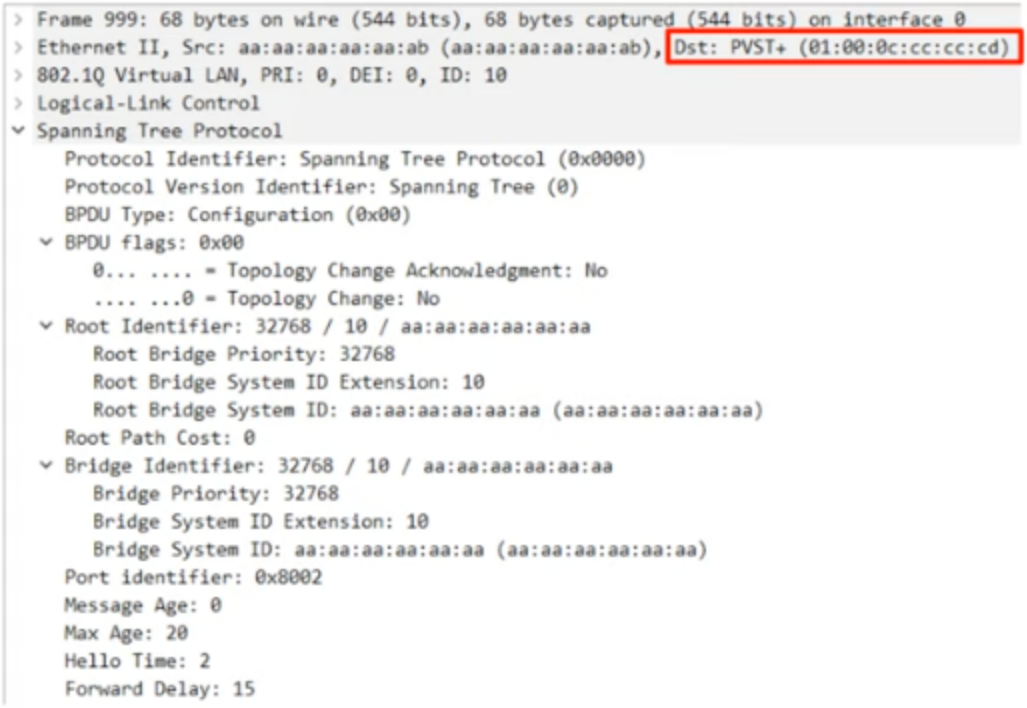
- **What happens next:** After the timer reaches zero, the switch assumes the path to the root is lost. It will immediately **reevaluate its STP choices** (root bridge, root port, designated ports, etc.).
- **The Transition:** If, after its re-evaluation, the switch decides a different port (which was previously in a **Blocking** state) should now become a Root or Designated Port, that port will begin its transition to Forwarding.
- **Full Timeline for a Blocking Port:**
    1. **Max Age:** 20 seconds (waiting to confirm the failure).
    2. **Listening:** 15 seconds (transitional state).
    3. **Learning:** 15 seconds (transitional state).
    4. **Total:** Up to **50 seconds** for a blocking port to fully transition to forwarding after a link failure.

STP BPDU (Bridge Protocol Data Unit)

The BPDU is a special frame used by STP to exchange information. Cisco's PVST+ uses a **destination MAC address** of `0100.0ccc.cccd` (while standard IEEE STP uses `0180.c200.0000`).

The BPDU payload includes:

- PVST = Only ISLL Trunk encapsulation
- PVST+ = Supports 802.1Q



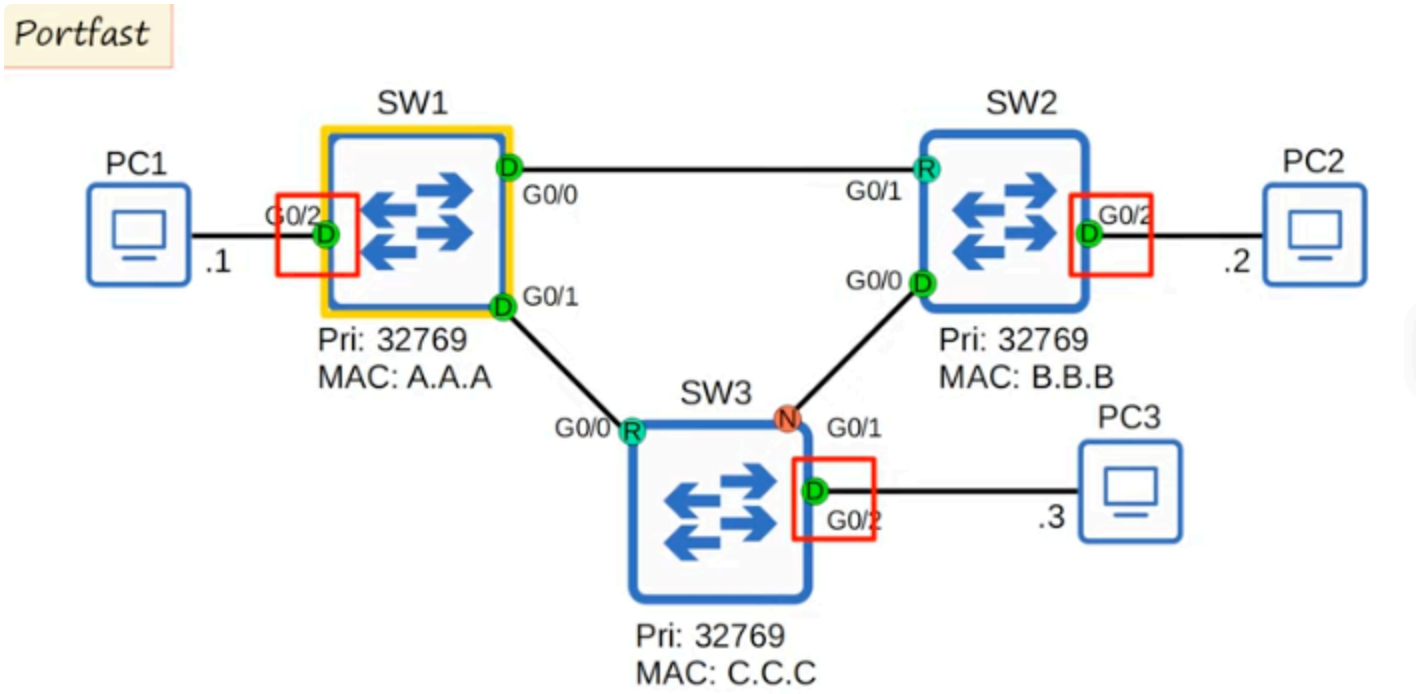
- **Protocol Identifier**: `0x0000` for STP.
- **Protocol Version Identifier**: `0` for classic STP (different for RSTP).
- **BPDU Type**: `0x00` for configuration BPDU.
- **Flags**: Used for topology change notification.
- **Root Identifier**: Contains bridge priority + extended system ID (VLAN ID) + MAC address of the root bridge.
- **Root Path Cost**: Cost from this bridge to the root (0 if this is the root).
- **Bridge Identifier**: Same structure as root identifier – indicates the bridge sending the BPDU.
- **Port Identifier**: The interface that sent the BPDU – first half is port priority (default 128 decimal, `0x80`), second half is port number (e.g., `0x02`).
- **Timers**: Max Age, Hello, Forward Delay (sent by root bridge, overwritten by other switches).
- **Message Age**: Starts at 0 at root, incremented by 1 each time forwarded; subtracted from Max Age when received.

Optional STP Features (Spanning Tree Toolkit)

PortFast

PortFast is a Cisco proprietary spanning-tree enhancement that allows a switch port to bypass the **Listening** and **Learning** transitional states and move **immediately** to the **Forwarding** state. It is part of the "Spanning Tree Toolkit" and is designed to solve the 30-second delay experienced when connecting an end host to a switch port.

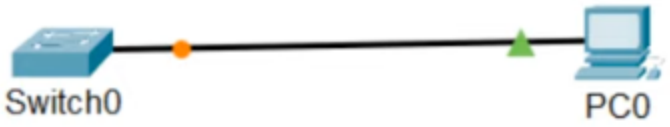
The Problem PortFast Solves



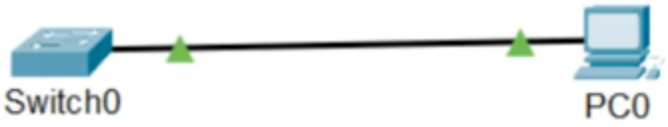
State	Default Duration	Behaviour
Listening	15 seconds	Port only processes BPDUs; no user traffic, no MAC learning
Learning	15 seconds	Port only processes BPDUs; **does** learn MAC addresses
Total Delay	**30 seconds**	Before the port can forward user traffic

When a PC, printer, server, or other end host is connected to a switch port, the port must pass through both transitional states before it can forward traffic. This results in a noticeable 30-second delay before the host can access the network.

Visual Behaviour (Packet Tracer / Real World)



1. **Connect a PC to a switch port.**
2. **Immediately:** The link light is **orange** (port is in Listening/Learning).
3. **After 30 seconds:** The link light turns **green** (port has entered Forwarding).



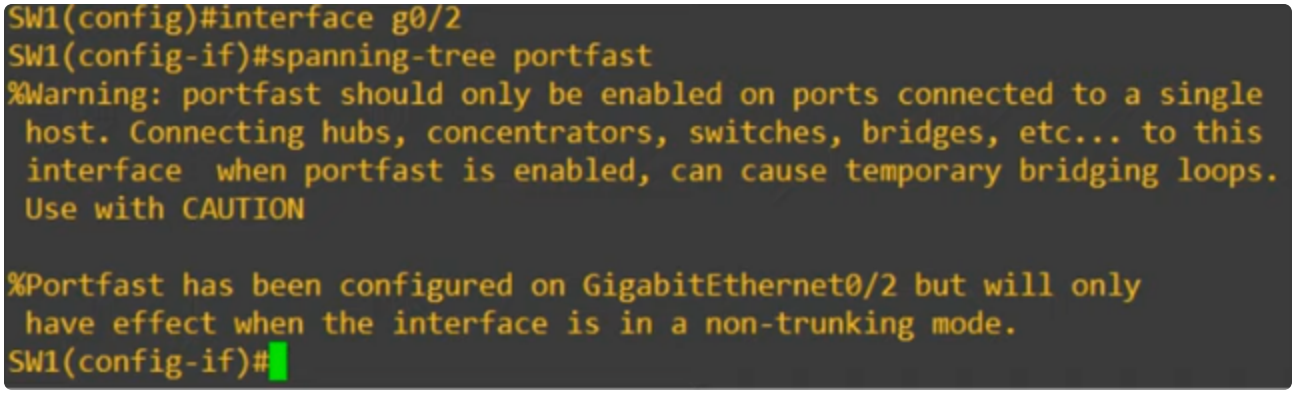
4. **With PortFast enabled:** The link light turns **green immediately** — no delay.

Why PortFast is Safe on End Hosts

- **Layer 2 loops** can only be formed when a switch is connected to another switch.
- An **end host** (PC, printer, IP phone, server) **cannot** create a Layer 2 loop because it does not forward frames.
- Therefore, there is **no risk** in bypassing the Listening and Learning states on ports connected to end hosts.

Configuration Commands

Interface-Level Configuration




```
interface GigabitEthernet0/2
 spanning-tree portfast
```

- A warning is displayed, reminding you to only enable PortFast on ports connected to end hosts.
- PortFast will **only take effect** if the interface is configured as an **access port** (non-trunking mode).
- Trunk ports are typically connected to other switches, so PortFast is automatically ignored on trunk ports.

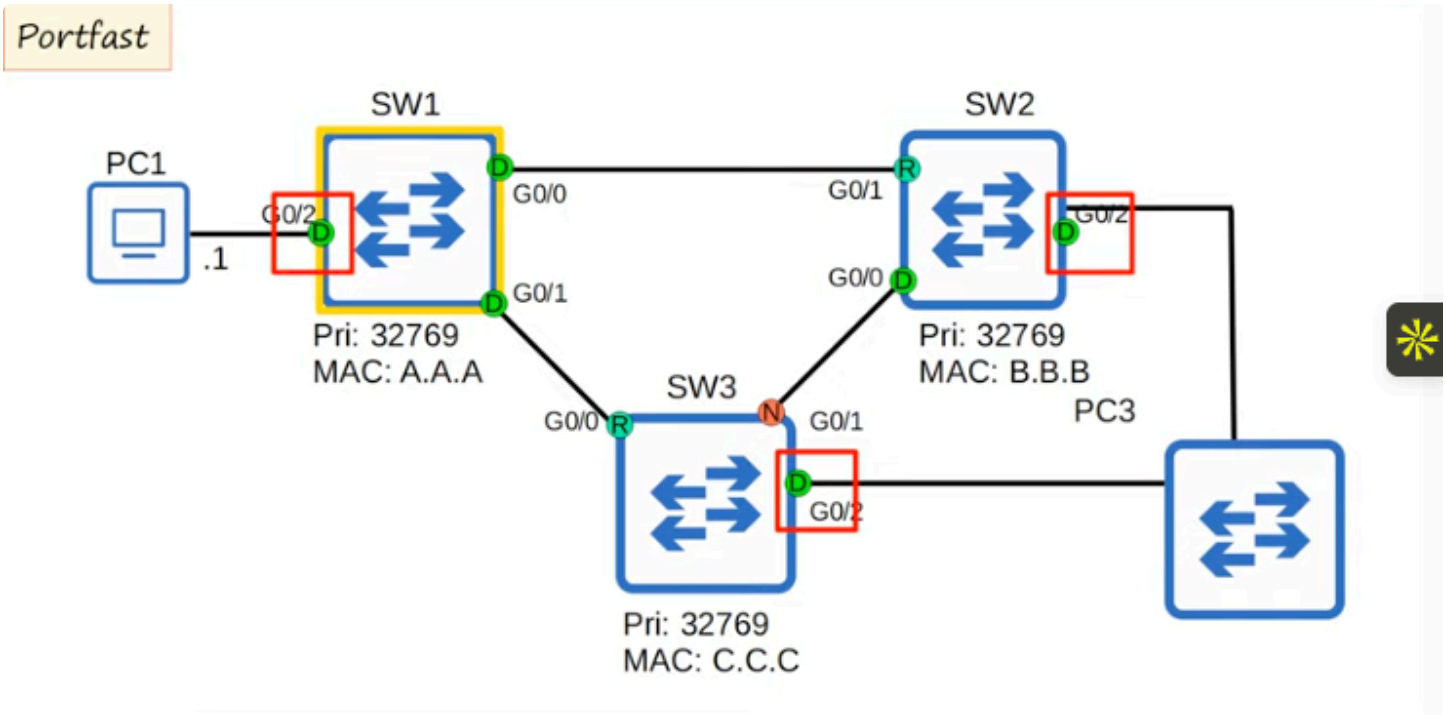
Global Configuration

```
spanning-tree portfast default
```

- Enables PortFast on **all access ports** on the switch.
- Does **not** affect trunk ports.
- This is the preferred method when deploying PortFast across many interfaces.

Risks and Limitations

Key Risk: Accidental Switch Connection



- If an employee (unknowingly or accidentally) connects another switch to a PortFast-enabled port, a **Layer 2 loop** will form because the port skipped its loop-prevention checks.
- Example scenario: A user moves a PC and plugs the cable into a different switch port, but inadvertently connects another switch to the original PortFast-enabled port.

Best Practice: Always pair PortFast with BPDU Guard on end-host ports. BPDU Guard will disable the port if a BPDU is received, preventing accidental loops from rogue switch connections.

BPDU Guard

BPDU Guard is a spanning-tree security feature designed to protect against Layer 2 loops caused by the accidental or malicious connection of a switch to a PortFast-enabled port. When a BPDU is received on a BPDU Guard-enabled interface, the port is immediately placed into an **err-disabled** (shutdown) state, preventing any potential loop from forming.

The Problem BPDU Guard Solves

- **PortFast** bypasses the Listening and Learning states, meaning a port can begin forwarding immediately.
- If a switch is accidentally connected to a PortFast-enabled port, a **Layer 2 loop** can form instantly because the port did not go through its normal loop-prevention checks.

- **BPDU Guard** acts as a safety net: if any BPDU (indicating another switch) is detected on the port, the port is disabled before a loop can occur.

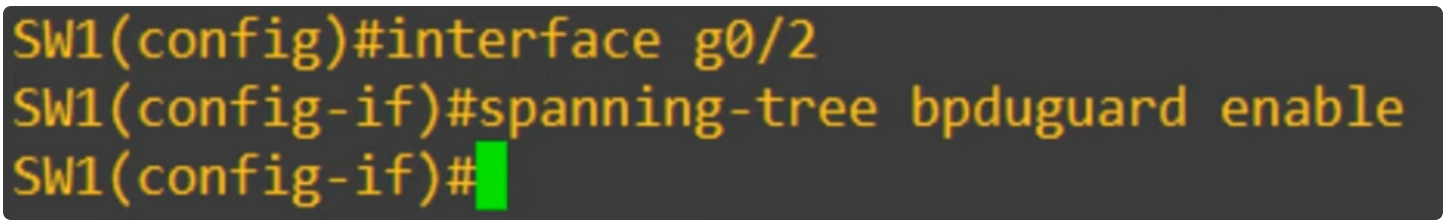
Use Case

BPDU Guard is **typically enabled on PortFast-enabled ports** connected to end hosts. This combination is considered a best practice:

Feature	Role
PortFast	Allows immediate forwarding (bypasses Listening/Learning)
BPDU Guard	Disables the port if a switch is detected (prevents loops)

Configuration Commands

Interface-Level Configuration




```
interface GigabitEthernet0/2
 spanning-tree bpduguard enable
```

- Enables BPDU Guard on a **single interface**.
- No mention of PortFast in the command — it can be applied independently.
- The interface will be disabled if any BPDU is received.

Global Configuration

```
spanning-tree portfast bpduguard default
```

- Enables BPDU Guard on **all PortFast-enabled interfaces** across the switch.
- **Note the syntax:** The global command **must** include `portfast` in the command string.
- Only takes effect on interfaces that already have PortFast enabled (either individually or via `spanning-tree portfast default`).

Command Syntax Comparison

Level	Command	Scope
Interface	`spanning-tree bpduguard enable`	Single interface only
Global	`spanning-tree portfast bpduguard default`	All PortFast-enabled interfaces

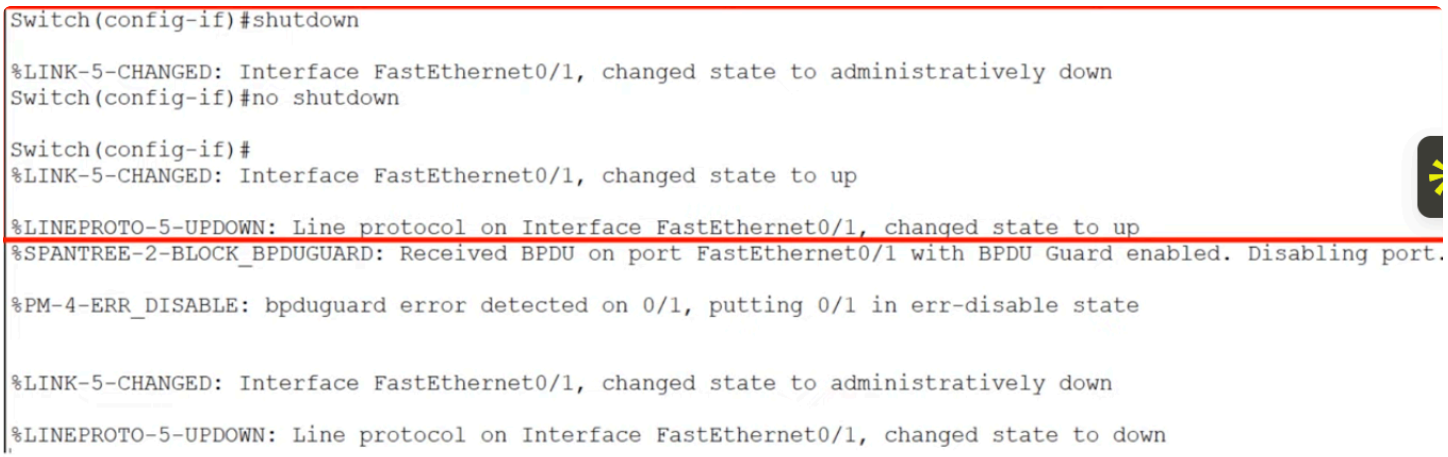
Behaviour When a BPDU is Received

```
%SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port FastEthernet0/1 with BPDU Guard
enabled. Disabling port.

%PM-4-ERR_DISABLE: bpduguard error detected on 0/1, putting 0/1 in err-disable state


%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
```

1. A BPDU arrives on a BPDU Guard-enabled interface.
2. The switch immediately places the port into an **err-disabled** (shutdown) state.
3. The port stops forwarding all traffic — **no loop can form**.

Recovery Procedure

To re-enable a port that was disabled by BPDU Guard:




```
interface GigabitEthernet0/2
 shutdown
 no shutdown
```

Important Caveat

- If the **root cause** is not resolved (e.g., another switch is still connected to the port), the port will be **immediately disabled again** when the next BPDU is received.
- **Always resolve the underlying problem** before attempting to re-enable the interface.

Root Guard

Root Guard	If you enable **root guard** on an interface, even if it receives a superior BPDU (lower bridge ID) on that interface, the switch will not accept the new switch as the root bridge. The interface will be disabled.

- **Purpose**: If a superior BPDU (lower bridge ID) is received on a Root Guard-enabled interface, the interface is disabled. This prevents an unauthorized switch from becoming the root bridge.
- **Use case**: Protects the designated root bridge position.

Loop Guard

Loop Guard	If you enable **loop guard** on an interface, even if the interface stops receiving BPDUs, it will not start forwarding. The interface will be disabled.

- **Purpose**: If a port stops receiving BPDUs (e.g., due to a unidirectional link failure), Loop Guard prevents it from transitioning to forwarding. The port is disabled instead.
- **Use case**: Prevents loops caused by unidirectional links.

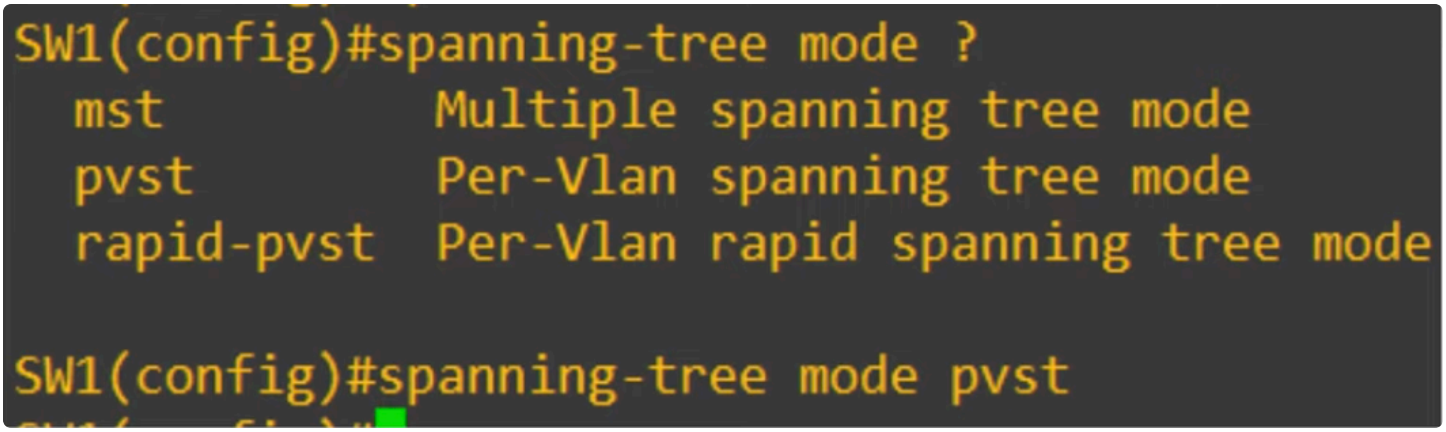
For the CCNA, PortFast and BPDU Guard are the most important optional features. Root Guard and Loop Guard should be known at a basic level.

Spanning Tree Configuration

STP runs by default, but administrative changes can be made to control the root bridge, port costs, and priorities.

Configuring the STP Mode

Modern Cisco switches run **Rapid-PVST** by default. To change to classic PVST (Per-VLAN Spanning Tree with Cisco extensions):

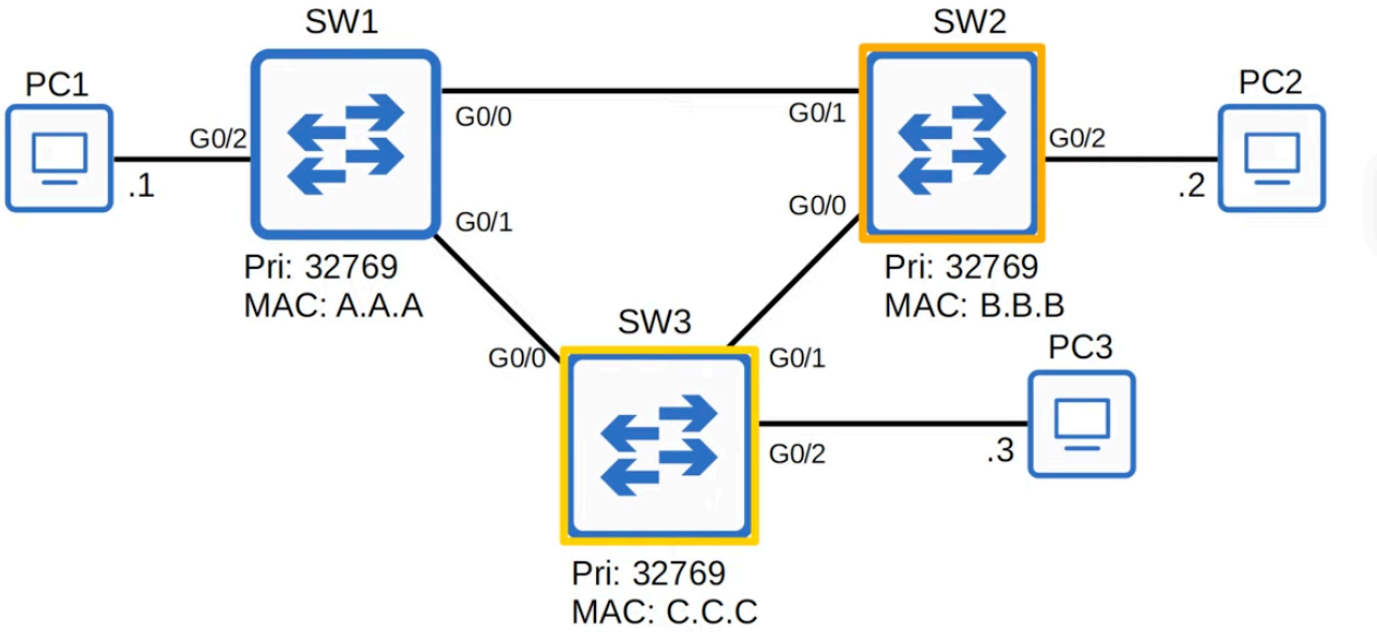



```
spanning-tree mode pvst
```

(Other modes: `mst` – multiple spanning tree, and `rapid-pvst`.)

Configuring the Root Bridge

The root bridge can be set manually using the `spanning-tree vlan VLAN-ID root primary` or `root secondary` commands. These automatically adjust the bridge priority.



SW1 is the root bridge, However we can configure SW3 to be the root bridge. We can also configure something called the secondary root bridge which will be next in line to become the root bridge if the current root bridge fails.

```
SW3(config)#spanning-tree vlan 1 root primary
SW3(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     cccc.cccc.cccc
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1)
              Address     cccc.cccc.cccc
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec
```

- `spanning-tree vlan 1 root primary` → sets priority to **24576** (or lower if another bridge already has a lower priority lower than 24576, it sets this switch priority to 4096 less than the other switch's priority).

```
SW2(config)#spanning-tree vlan 1 root secondary
SW2(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     cccc.cccc.cccc
              Cost        4
              Port        1 (GigabitEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28673  (priority 28672 sys-id-ext 1)
              Address     bbbb.bbbb.bbbb
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec
```

- `spanning-tree vlan 1 root secondary` → sets priority to **28672** (makes it second-lowest, so it becomes root if the primary fails).

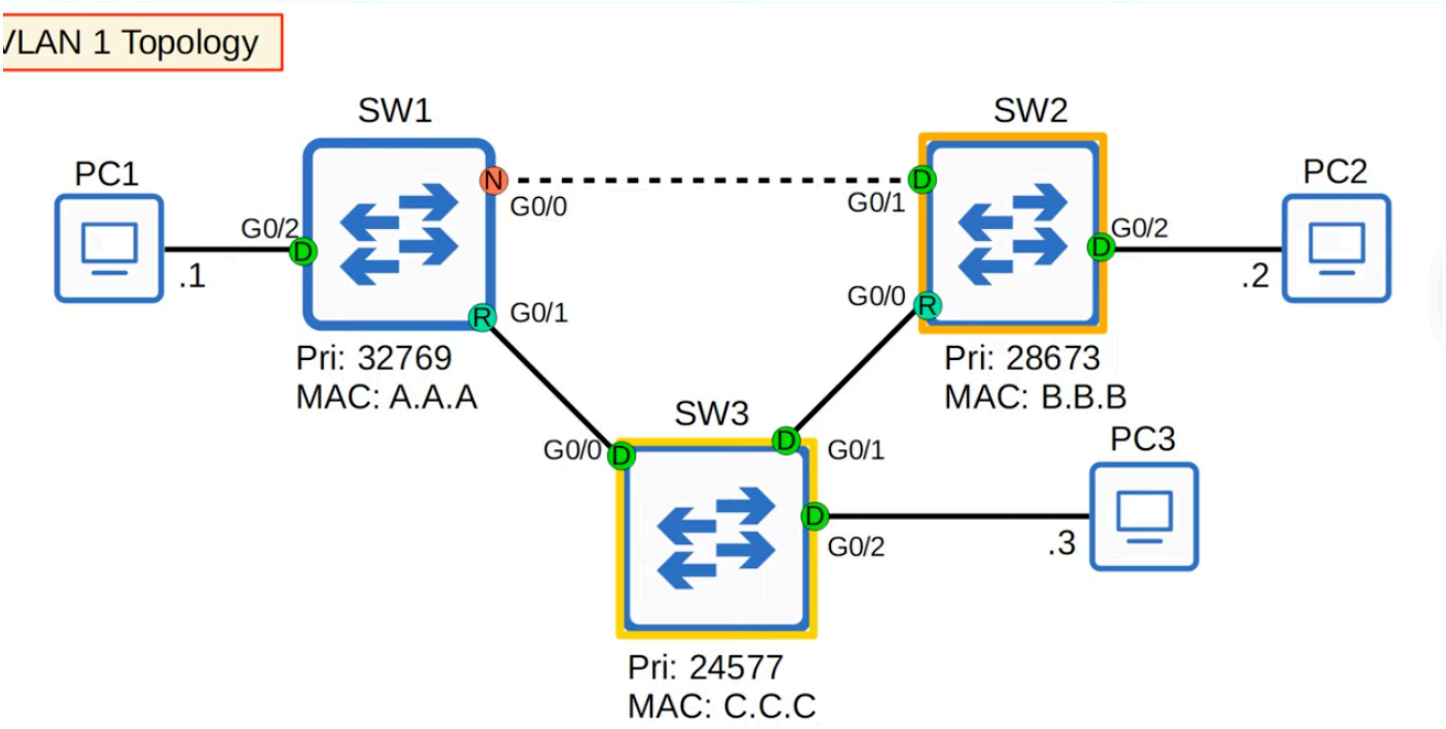
Alternatively, the priority can be set directly (in increments of 4096):

```
spanning-tree vlan 1 priority 24576
```

Load Balancing Across VLANs

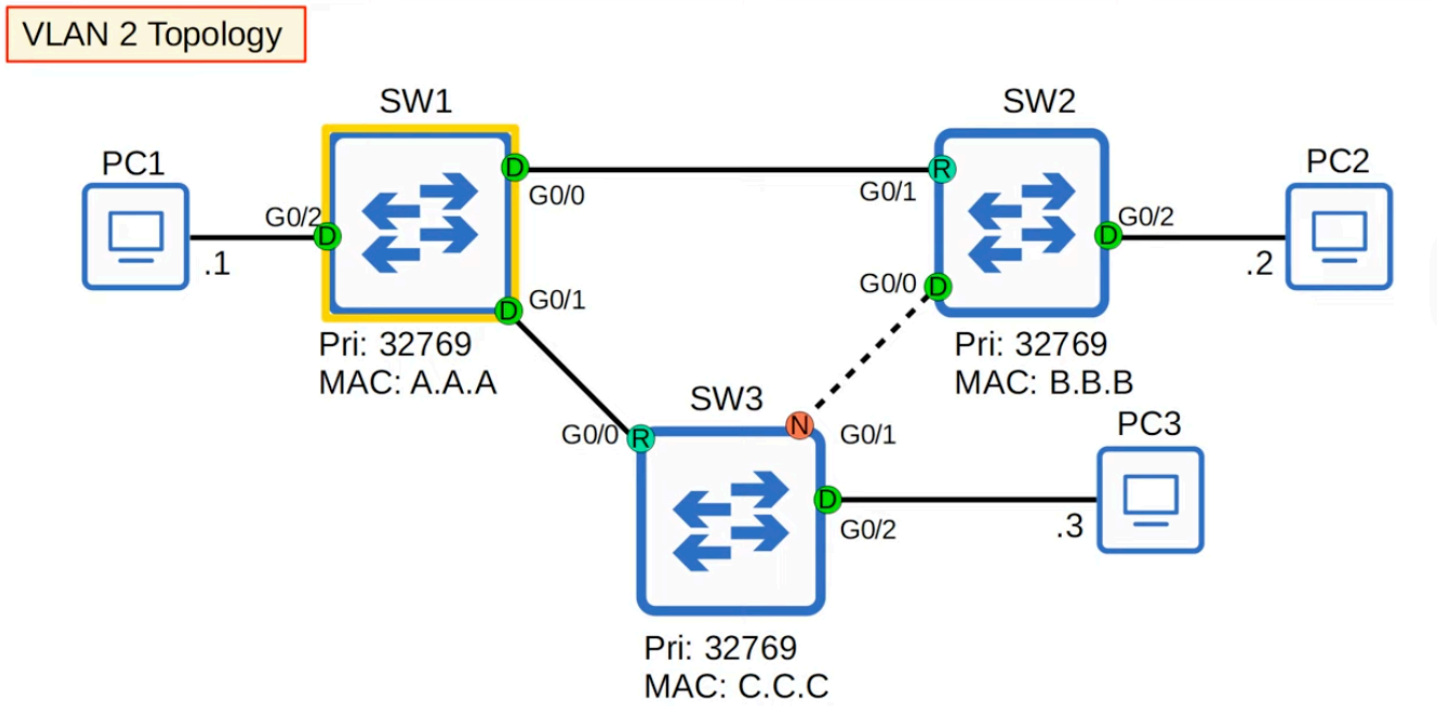
Context Update: Default Topology Applied to VLAN 1

The speaker begins by showing the current converged topology **for VLAN 1**:



- **Interface G0/0 on SW1 is blocking** (the connection between SW1 and SW2 is disabled).
- This is the result of the root bridge configuration applied **only to VLAN 1**.
- The network is running Cisco **PVST+** (Per-VLAN Spanning Tree Plus).

The Problem: The Same Topology for VLAN 2



The speaker then asks: *What will the topology look like for VLAN 2?*

Answer: The default topology, because the root bridge settings configured **only apply to VLAN 1**.

- In **VLAN 2**, the connection between SW1 and SW2 will **NOT** be disabled.
- Instead, the connection between **SW2 and SW3** will be disabled.

Visual Comparison

Topology	VLAN 1	VLAN 2
Blocked Link	SW1 ↔ SW2	SW2 ↔ SW3
Root Bridge	Custom (configured for VLAN 1)	Default root

The Concept: Spanning Tree Load Balancing

The speaker introduces **load balancing** as a key benefit of PVST+.

Why Load Balancing Matters

- If you have **multiple VLANs** in your network, blocking the **same interface** in each VLAN is a **waste of interface bandwidth**.
- That blocked connection will sit idle, doing nothing except waiting for another link to fail so it can begin forwarding.
- This is inefficient — the redundant bandwidth is effectively unused.

The Solution

- Configure a **different root bridge** for different VLANs.
- Result: **Different VLANs will block different interfaces**.
- This allows traffic from different VLANs to use different redundant links simultaneously.

Key Takeaway (from speaker): "If you configure a different root bridge for different VLANs, different VLANs will disable different interfaces."

Summary of Load Balancing Benefits

Before Load Balancing	After Load Balancing
Same link blocked for ALL VLANs	Different links blocked per VLAN
One redundant link sits completely idle	Both redundant links carry traffic
Wasted bandwidth	Bandwidth is utilized efficiently

Configuration Principle

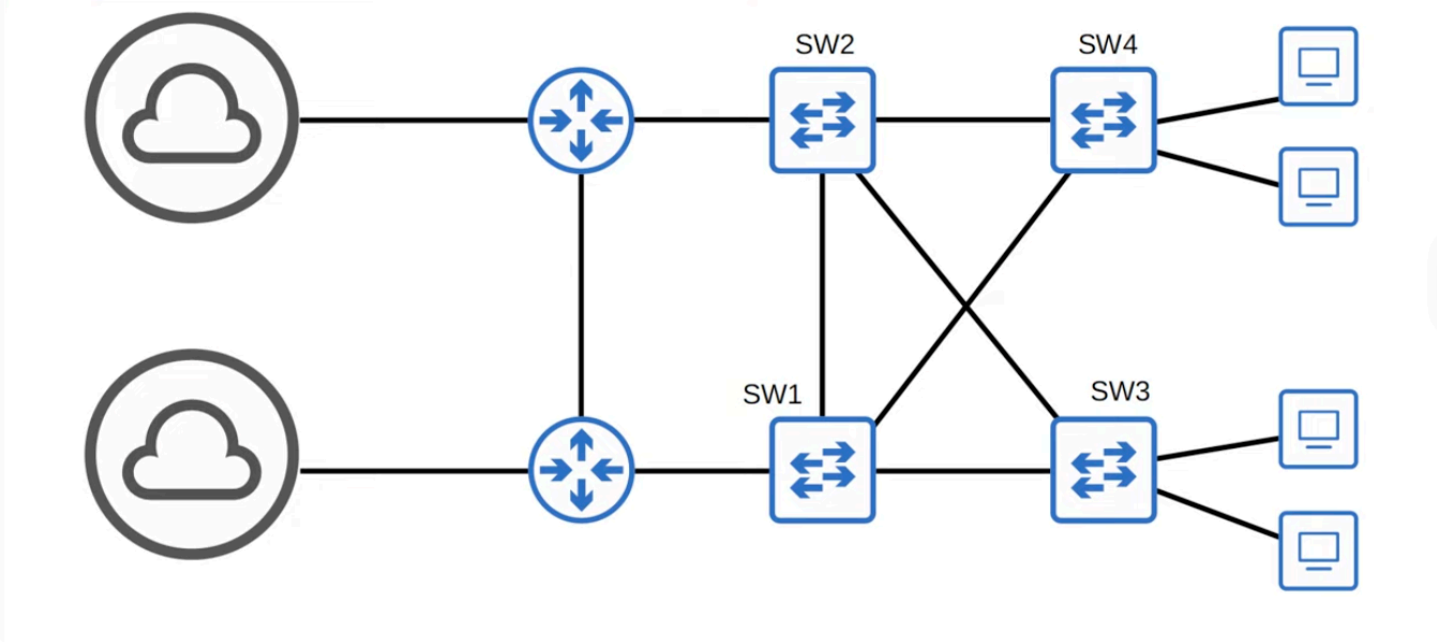
To achieve load balancing across VLANs:

VLAN	Primary Root Bridge	Resulting Blocked Link
VLAN 1	SW1 (configured)	SW1 ↔ SW2
VLAN 2	Default root (e.g., SW3)	SW2 ↔ SW3

> ***Professional Note: The speaker emphasizes that the root bridge configuration applied only to VLAN 1 in the example. For VLAN 2, the default STP election process determines the root bridge, resulting in a different blocked interface. This demonstrates how PVST+ enables per-VLAN path optimization.***

Quiz Review: Per-VLAN Root Bridge Configuration for Load Balancing

Quiz Question



Scenario: Two VLANs are active in the network — VLAN 10 and VLAN 20. By default, SW3 is the root bridge for both VLANs.

Objective:

- Configure **SW1** as the **primary root** for VLAN 10 and the **secondary root** for VLAN 20.
- Configure **SW2** as the **primary root** for VLAN 20 and the **secondary root** for VLAN 10.

On SW1 (Commands)

```
spanning-tree vlan 10 root primary
spanning-tree vlan 20 root secondary
```

- **VLAN 10:** SW1 becomes the root bridge (priority set to 24576).
- **VLAN 20:** SW1 becomes the backup root bridge (priority set to 28672). It will become root only if the primary (SW2) fails.

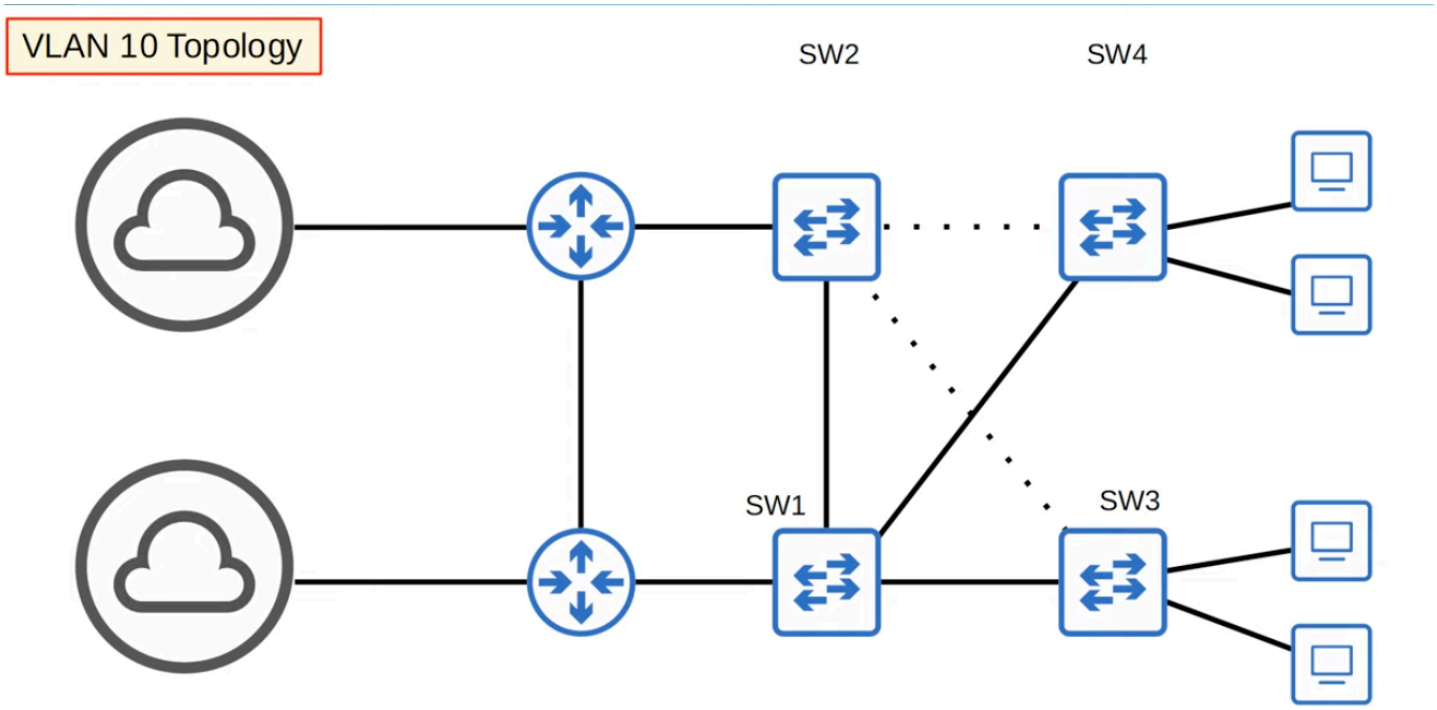
On SW2 (Commands)

```
spanning-tree vlan 20 root primary
spanning-tree vlan 10 root secondary
```

- **VLAN 20:** SW2 becomes the root bridge (priority set to 24576).
- **VLAN 10:** SW2 becomes the backup root bridge (priority set to 28672). It will become root only if the primary (SW1) fails.

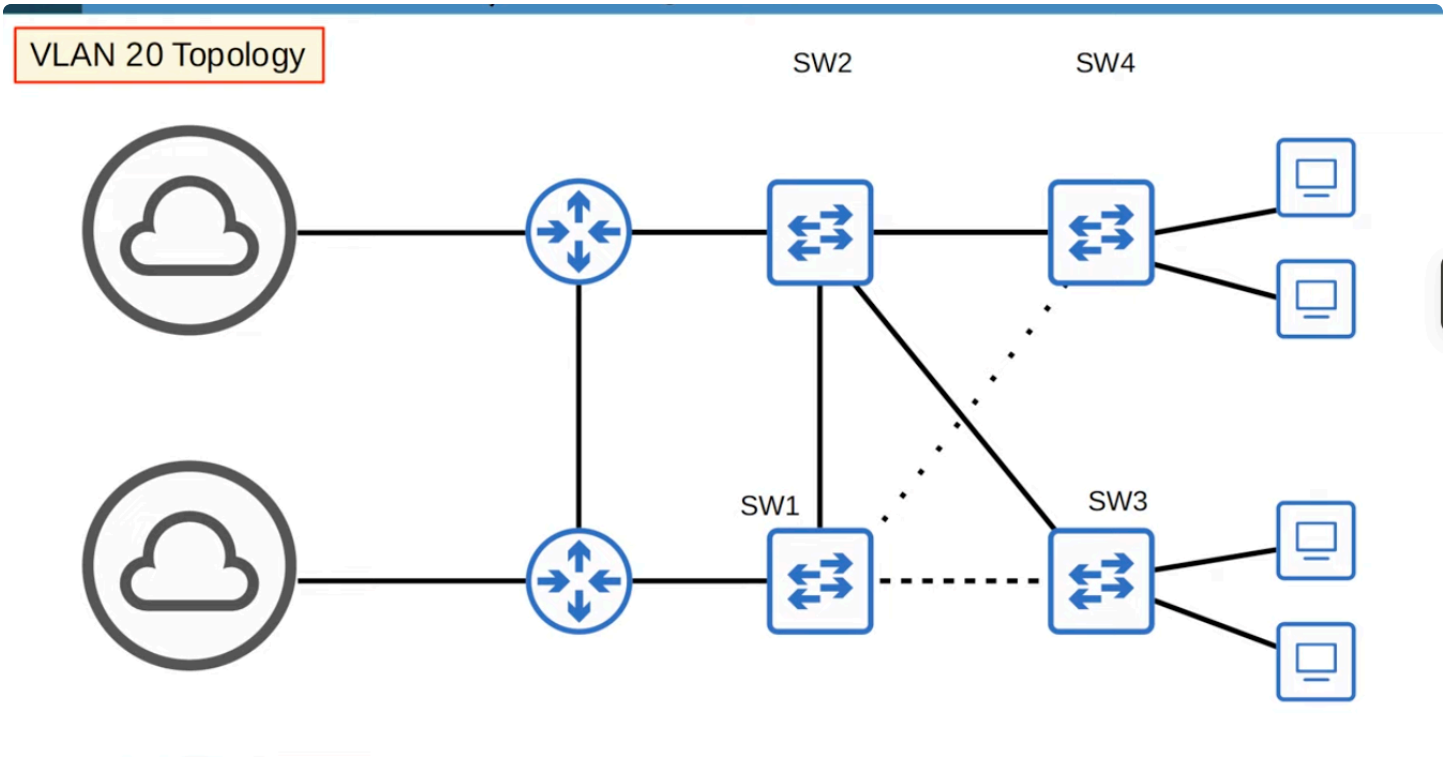
Resulting Topology

VLAN 10 Topology



- **Root Bridge:** SW1
- **Blocked Link:** (Determined by STP calculation with SW1 as root)

VLAN 20 Topology



- **Root Bridge:** SW2
- **Blocked Link:** (Determined by STP calculation with SW2 as root)

> ***Key Insight: Because different switches are root for different VLANs, different interfaces will be blocked in each VLAN. This achieves load balancing — both redundant links carry traffic instead of one sitting idle.***

Command Summary Table

Switch	VLAN 10	VLAN 20
SW1	`spanning-tree vlan 10 root primary`	`spanning-tree vlan 20 root secondary`
SW2	`spanning-tree vlan 10 root secondary`	`spanning-tree vlan 20 root primary`

Resulting Bridge Priorities

Switch	VLAN 10 Priority	VLAN 20 Priority
SW1	24576 (Primary)	28672 (Secondary)
SW2	28672 (Secondary)	24576 (Primary)
SW3	32768 (Default)	32768 (Default)
SW4	32768 (Default)	32768 (Default)

Key Takeaway

By configuring **different root bridges for different VLANs**, the network achieves **spanning tree load balancing** — traffic from each VLAN uses different redundant links, eliminating wasted bandwidth.

STP Port Configuration

Overview

Spanning Tree Protocol allows administrators to manually configure two port-level settings — **cost** and **priority** — to influence the root port and designated port selection process. Both settings are configured on a **per-VLAN basis**, similar to bridge priority.

```
SW2(config-if)#spanning-tree vlan 1 ?
  cost           Change an interface's per VLAN spanning tree path cost
  port-priority  Change an interface's spanning tree port priority

SW2(config-if)#spanning-tree vlan 1
```

Port Cost Definition

The **cost** of an interface represents the path cost to the root bridge. It is the primary factor used to determine the **root port** and serves as a tiebreaker when selecting designated and non-designated ports.

Default Cost Values (Review from Day 20)

Interface Speed	STP Cost
10 Mbps (Ethernet)	100
100 Mbps (FastEthernet)	19
1 Gbps (GigabitEthernet)	4
10 Gbps (10 GigabitEthernet)	2

Configuration

- **Command:** `spanning-tree vlan VLAN-ID cost VALUE`
- **Range:** 1 to 200,000,000

```
SW2(config-if)#spanning-tree vlan 1 cost ?
  <1-200000000>  Change an interface's per VLAN spanning tree path cost

SW2(config-if)#spanning-tree vlan 1 cost 200
```

Port Priority Definition

The **port priority** forms the first half of the port ID. It is used as the **final tiebreaker** when determining the root port (after cost and bridge ID comparisons).

Configuration

- **Command:** `spanning-tree vlan VLAN-ID port-priority VALUE`
- **Range:** 0 to 224 (in increments of 32)
- **Default:** 128

```
SW2(config-if)#spanning-tree vlan 1 cost ?
  <1-200000000>  Change an interface's per VLAN spanning tree path cost

SW2(config-if)#spanning-tree vlan 1 cost 200
SW2(config-if)#spanning-tree vlan 1 port-priority ?
  <0-224>  port priority in increments of 32

SW2(config-if)#spanning-tree vlan 1 port-priority 32
SW2(config-if)#
```

Valid Priority Values (Increments of 32)

Decimal	Hex Equivalent
0	0x00
32	0x20
64	0x40
96	0x60
128	0x80 (Default)
160	0xA0
192	0xC0
224	0xE0

Why Change These Values?

Setting	Purpose
Cost	Influence which port becomes the root port by making a specific path appear cheaper or more expensive
Priority	Influence the final tiebreaker in root port selection when costs are equal

Both settings are used to **manually override** the default STP election results, allowing the administrator to force traffic along a desired path.

Command Syntax Summary

Setting	Command	Range	Default
Cost	`spanning-tree vlan VLAN-ID cost VALUE`	1 – 200,000,000	Speed-based
Priority	`spanning-tree vlan VLAN-ID port-priority VALUE`	0 – 224 (×32)	128

Key Exam Points

- Both settings are configured **per-VLAN**, per-interface.
- **Cost** is the primary factor for root port selection.
- **Priority** is the final tiebreaker in root port selection.
- Port priority must be configured in **increments of 32** (values 0, 32, 64, 96, 128, 160, 192, 224).
- The default port priority is **128** (hex 0x80).