Spanning Tree Protocol (STP) Workbook

A Hands-On Guide to PVST+, RSTP, and MSTP

Mehdi JAFARI ZADEH

Date: February 5, 2025

Contents

Chapter 1

Spanning Tree Protocol (STP)

STP Introduction

Spanning Tree Protocol (STP) is essential for preventing broadcast storms and loops in switched networks with redundant paths. In this section, we'll cover the purpose of STP, its importance in maintaining a loop-free topology, and an overview of how it dynamically reconfigures the network when changes occur. This foundation sets the stage for understanding how STP supports network reliability and resilience.

Root Bridge

The Root Bridge is the central reference point in the STP topology. It is elected based on the lowest bridge ID, which is a combination of the bridge priority and MAC address. All other switches in the network compute the best path back to the Root Bridge. Understanding how the Root Bridge is selected is crucial because it directly influences the network's spanning tree structure and the flow of data through the network.

STP Root Port, Designated Port

Within each switch (except for the Root Bridge), the Root Port is the interface that offers the best path to the Root Bridge, based on the lowest path cost. On each network segment, the Designated Port is the switch port that has the best path to the Root Bridge and is responsible for forwarding traffic toward it. Together, these port roles determine the active topology of the network by ensuring there is only one active path between any two network devices, thereby preventing loops.

Network Topology Overview

In this exercise, you will work with the network topology as described below. The switches (SW1 to SW6) are connected as Figure 1.1.

The MAC addresses of the switches are as follows:

$$\text{MAC SW1} < \text{MAC SW2} < \text{MAC SW3} < \text{MAC SW4} < \text{MAC SW6} < \text{MAC SW6}$$

This means **SW1** has the lowest MAC address and **SW6** has the highest.

Step-by-Step Tasks

1. **Switch STP Mode to Traditional STP**
   Change the STP mode on all switches to **traditional STP (IEEE 802.1D)**.

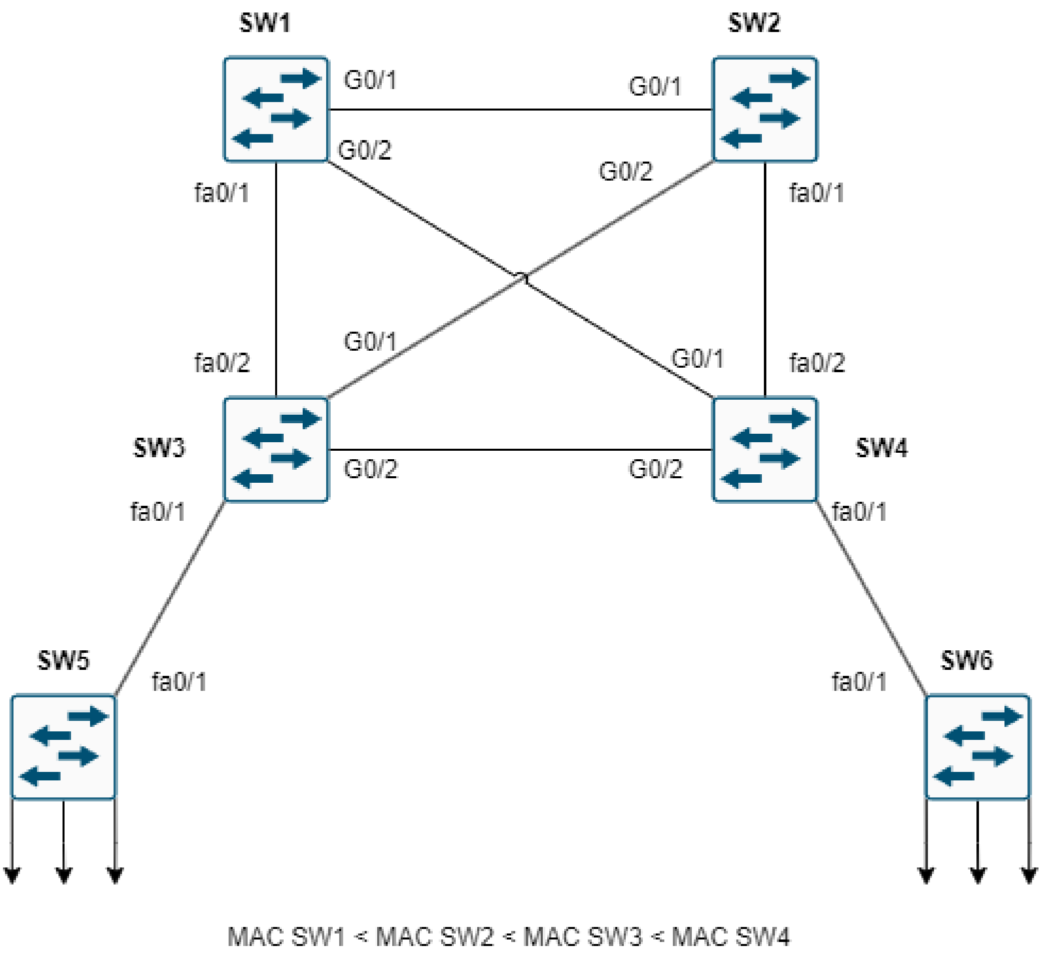

Figure 1.1:

2. **Disable Unused Interfaces**
   Turn off any extra or unused interfaces to prevent unnecessary loops.

3. **Set Links to Point-to-Point Mode**
   Configure all links between switches as **Point-to-Point (P2P)** to optimize STP convergence.

4. **Verify Bridge IDs**
   Check the **Bridge ID** for each switch to see how STP will select the Root Bridge.

5. **Predict the Root Bridge**
   Based on the MAC addresses provided (**SW1** < **SW2** < **SW3** < **SW4** < **SW5** < **SW6**), estimate which switch will become the Root Bridge.

   - *Question:* Which switch is likely to be the Root Bridge and why?

6. **Change the Root Bridge by Adjusting Priority**
   Modify the **priority** settings so that the switch identified in the diagram becomes the Root Bridge.

   - *Note:* Lower priority values increase the chance of a switch becoming the Root Bridge.

7. **Calculate Path Costs for Active Ports**
   For each non-root switch, **calculate the cost** of paths to the Root Bridge based on link speeds. Use standard STP cost values:

   - Gigabit Ethernet (1 Gbps) = 4

- Fast Ethernet (100 Mbps) = 19

8. **Estimate Root Ports**
   Identify the **Root Port** on each non-root switch. This is the port with the lowest path cost to the Root Bridge.

   - *Question:* Which port will be the Root Port on SW2, SW3, SW4, SW5, and SW6?

9. **How Many Root Bridges Are There in a Network?**

   - *Question:* In any given STP-enabled network, how many Root Bridges can exist?

10. **How Many Root Ports Are There in a Network?**

    - *Question:* In any given STP-enabled network, how many Root Port can exist?

11. **How Many Designated Ports (DP) Are in the Topology?**

    - *Question:* In any STP-enabled topology, how many Designated Ports (DP) will there be?

12. **Estimate the Designated Ports on Each Switch**

    - Before using any commands, predict which ports will be Designated Ports on each switch in the topology.

13. **Verify Designated Ports Using Commands**

14. **Are All Root Bridge Ports Designated Ports?**

    - *Question:* Based on your observations, can you conclude that all ports on the Root Bridge are Designated Ports?

15. **Change the Port Cost of G0/2 on SW4**

    - Modify the path cost of interface G0/2 on SW4 to 15. This will influence STP's decision on which path to use.

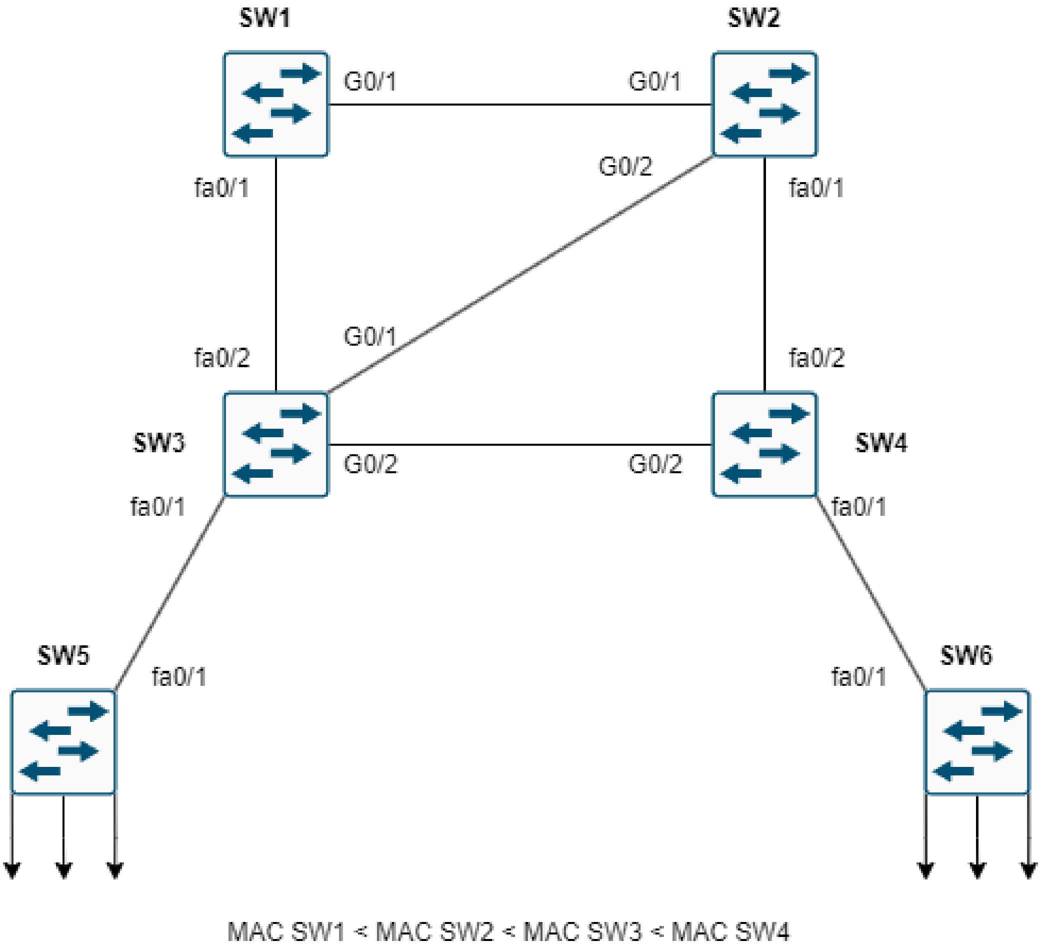

Figure 1.2:

16. **Predict the New Root Port on SW4 After Link Removal**

    - If you remove the link connected to G0/1 on SW4 (Figure 1.2), which port will become the new Root Port? Explain your reasoning based on path costs and STP rules.
    - *Hint:* Consider which remaining port on SW4 has the lowest cost path to the Root Bridge.

17. **Identify Root Ports with SW1 as the Root Bridge**

    - In this topology, assuming SW1 is the Root Bridge, identify the Root Ports on the other switches (SW2, SW3, SW4, SW5, and SW6).

18. **Change the Port ID to Influence Link Selection**

    - STP uses Port IDs as a tie-breaker when path costs are equal. By adjusting the Port Priority (which is part of the Port ID), you can influence which link STP selects as the active path.
    - *Task:* Change the Port Priority of specific interfaces to modify STP's link selection.

Additional exercises

Draw the Active Topology for Each Scenario

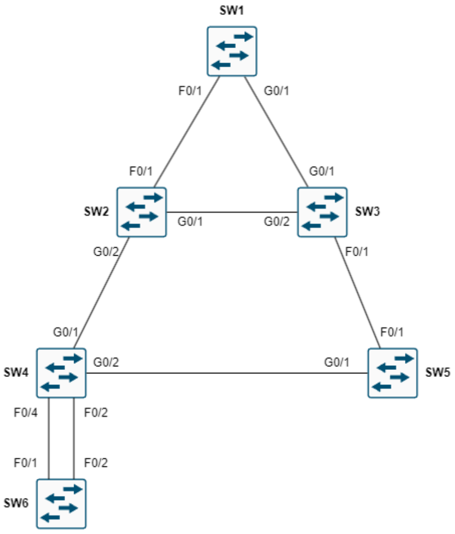

Figure 1.3:

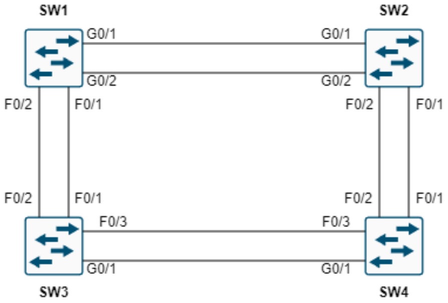

Figure 1.4:

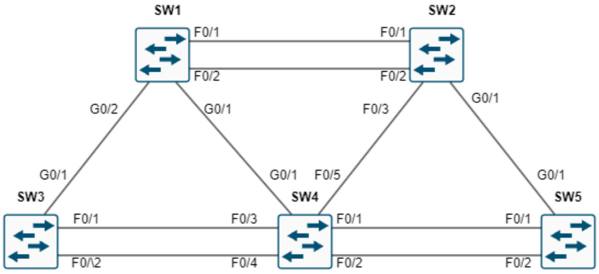

Figure 1.5:

Chapter 2

PVST (Per-VLAN Spanning Tree)

What is PVST?

Per-VLAN Spanning Tree (PVST) is a version of the Spanning Tree Protocol (STP) that creates a separate spanning tree instance for each VLAN in the network. This allows for better load balancing and more efficient use of network resources, as different VLANs can have different root bridges and forwarding paths.

Key Features of PVST:

1. **Separate Spanning Tree per VLAN:**
   Each VLAN runs its own independent instance of STP, meaning the topology can be optimized differently for each VLAN.

2. **Improved Load Balancing:**

   By assigning different root bridges to different VLANs, PVST allows traffic to be distributed across multiple switches, reducing congestion and improving performance.

3. Root Bridge Election per VLANCompatibility with Cisco Devices:

   VST is a Cisco proprietary protocol and is typically used in Cisco environments. It is based on the original IEEE 802.1D standard but with VLAN-specific enhancements.

4. **Uses Common STP Concepts:**

   PVST uses the same roles and states as traditional STP:

   - Root Bridge
   - Root Port (RP)
   - Designated Port (DP)
   - Blocked Port

How PVST Works:

- **Root Bridge Election per VLAN:**

  Each VLAN elects its own Root Bridge based on the lowest Bridge ID (priority + MAC address). This allows different switches to act as the Root Bridge for different VLANs.

- **Port Roles and Path Selection:**

  For each VLAN, PVST determines the Root Ports and Designated Ports independently, which results in different forwarding paths for different VLANs.

Benefits of PVST:

- **Flexibility:** Allows network administrators to optimize traffic flow for each VLAN.
- **Load Balancing:** Distributes traffic across multiple links by having different root bridges for different VLANs.
- **Faster Convergence:** PVST+ (an enhanced version) offers faster convergence times compared to traditional STP.

Chapter 3

PVST+ Lab Exercise: VLAN-Based Root Bridge Configuration

Network Topology Overview

In this scenario, you will configure **PVST+ (Per-VLAN Spanning Tree Plus)**, which allows separate spanning tree instances for each VLAN. The network consists of three switches (**SW1**, **SW2**, and **SW3**) and multiple VLANs with end devices connected to each switch.

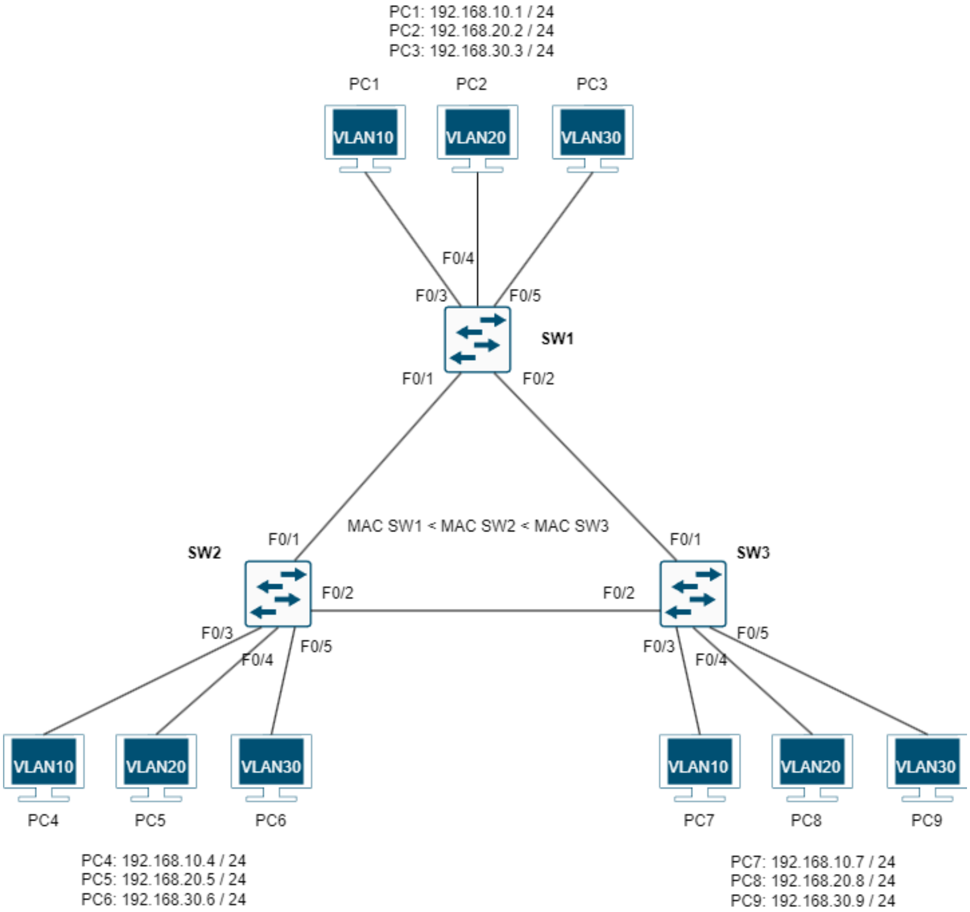

Figure 3.1:

- **MAC Address Order:**
  MAC SW1 < MAC SW2 < MAC SW3
  (SW1 has the lowest MAC address.)

- **VLANs and Devices:**
  - **VLAN 10:** PC1, PC4, PC7
  - **VLAN 20:** PC2, PC5, PC8
  - **VLAN 30:** PC3, PC6, PC9

Step-by-Step Configuration Tasks

1. **Set Inter-Switch Links to Trunk Mode**
   Configure the links between the switches (**SW1, SW2, SW3**) to operate in **trunk mode** to allow VLAN traffic to pass through.

2. **Verify Current STP Status**
   Before making further changes, **check the current STP status** to see how the network has chosen the Root Bridge for each VLAN.

3. **Enable VTP on All Switches**
   Configure **VTP (VLAN Trunking Protocol)** to manage VLANs centrally.

   VTP Settings:

   - **Domain Name:** `network`
   - **Version:** 2
   - **Password:** `CCNP`

   - SW1 Configuration (VTP Master/Server):

   - SW2 and SW3 Configuration (VTP Clients):

4. **Create VLANs 10, 20, and 30 on SW1 (VTP Server)**
   Since **SW1** is the VTP Server, create the VLANs here, and they will propagate to **SW2** and **SW3**.

5. **Verify STP Status After VLAN Creation**
   After setting up VTP and VLANs, check how **PVST** has adjusted the spanning tree roles.

PVST Root Bridge Configuration Per VLAN

1. **Set Root Bridges for Each VLAN Using PVST+**
   Configure different **Root Bridges** for each VLAN to optimize traffic flow:

   - **VLAN 10 Root Bridge:** Set **SW1** as the Root Bridge.
   - **VLAN 20 Root Bridge:** Set **SW2** as the Root Bridge.
   - **VLAN 30 Root Bridge:** Set **SW3** as the Root Bridge.

   - SW1 (Root for VLAN 10):

   - SW2 (Root for VLAN 20):

   - SW3 (Root for VLAN 30):

   Note: A lower priority value increases the likelihood of a switch becoming the Root Bridge. The default is 32768.

2. **Verify STP Status After Root Bridge Configuration**
   After configuring the Root Bridges for each VLAN, verify that the changes were applied correctly.

   Question:

   (a) Which switch is now the Root Bridge for each VLAN?

   (b) How did the Root Ports and Designated Ports change across the network?

3. **Draw the active topology of each violin.**
   - SW1 (Root for VLAN 10):
   - SW2 (Root for VLAN 20):
   - SW3 (Root for VLAN 30):

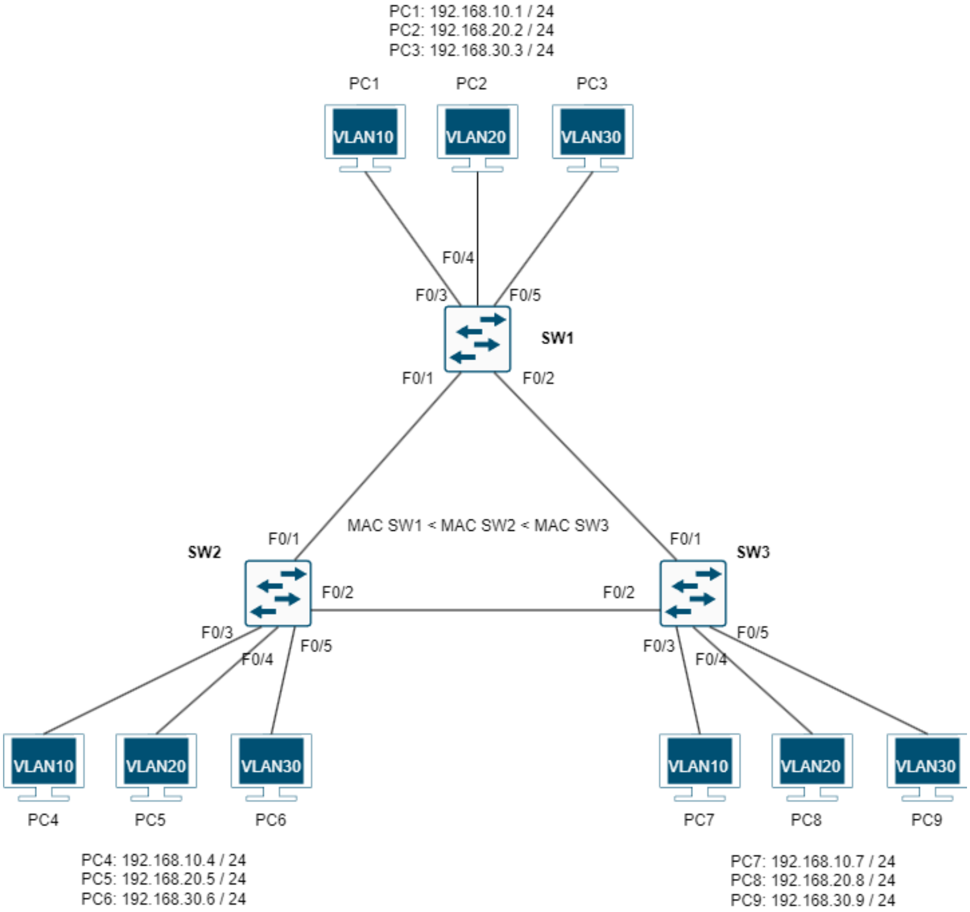

Figure 3.2:

Configure PortFast for End Devices

1. **Enable PortFast on All PC Ports**
   PortFast allows end devices (like PCs) to bypass STP's listening and learning states, enabling faster connections.

   Task: Configure **PortFast** on interfaces connected to PCs (e.g., **F0/3**, **F0/4**, **F0/5** on SW1).

2. **Verify Connectivity Between PC2 and PC3**
   Task: Test the connection between **PC2 (192.168.20.2)** and **PC3 (192.168.30.3)** using the **ping** command.

3. **Enable Debugging on SW3**
   Monitor STP behavior during topology changes.

   Task: Turn on STP debugging on **SW3**.

4. **Shut Down Interfaces to Trigger Topology Changes**
   Simulate link failures by shutting down specific interfaces:

   Task: Shut down **F0/1** on **SW3** and **F0/2** on **SW1**:

5. **Observe and Estimate New Topology**
   After shutting down the interfaces:

   Task: Check the **new port roles** and **STP state** using:

   Question: Which ports have become **Root Ports** or **Designated Ports**? Identify if any ports are now **Blocked**.

6. **Restore the Network to Its Original State**
   Re-enable the interfaces to bring the network back to its initial state.

   Task: Turn on **F0/1** on **SW3** and **F0/2** on **SW1**:

7. **Reduce Network Diameter for Faster Convergence**
   The **network diameter** defines the maximum number of switches a BPDU can pass through. Reducing it speeds up convergence.

   Task: Reduce the **network diameter** from the default **7** to **2**.

8. **Check Forward Delay and Max Age Timers**
   STP uses these timers to control how long switches stay in different states during convergence.

   Task: Verify **Forward Delay** and **Max Age** settings:

9. **Enable UplinkFast on SW2 for Faster Recovery**
   UplinkFast speeds up STP convergence when a primary link fails by immediately switching to a backup link.

   Task: Enable **UplinkFast** on **SW2**:

10. **Set Max-Update-Rate to 100 Packets per Second**
    Limit the number of update packets sent per second to prevent excessive flooding during topology changes.

    Task: Set **max-update-rate** to **100 packets/sec**:

11. **Verify Uplink Status on SW2**
    Task: Check if **UplinkFast** is working correctly:

Discussion Questions

1. How did reducing the network diameter impact convergence speed?

2. After enabling UplinkFast, how quickly did SW2 recover from link failures?

Chapter 4

Protecting the Spanning Tree Protocol (STP) Topology

The **Spanning Tree Protocol (STP)** is crucial for maintaining a loop-free and stable Layer 2 network topology. However, without proper safeguards, STP can become vulnerable to misconfigurations or malicious attacks, potentially leading to network loops, instability, or even downtime. To enhance the security and stability of the STP topology, network administrators can implement various protection mechanisms, including **BPDU Guard**, **Root Guard**, **BPDU Filter**, **Loop Guard**, and **UDLD (Unidirectional Link Detection)**. These features help prevent unauthorized devices from influencing STP operations, detect unidirectional links, and ensure consistent network behavior.

BPDU Guard

Bridge Protocol Data Units (BPDUs) are essential messages exchanged between switches to maintain the STP topology. **BPDU Guard** is a protective feature designed to secure **PortFast**-enabled ports, which are typically connected to end devices like computers or printers. Since these ports bypass STP's usual listening and learning states to achieve faster connectivity, receiving BPDUs on them could indicate a misconfiguration or an attack (such as connecting a rogue switch).

When **BPDU Guard** is enabled and a BPDU is received on a PortFast port, the switch immediately **disables** the port by placing it into an **err-disabled** (error-disabled) state. This action prevents potential network loops and unauthorized participation in the STP topology. BPDU Guard is particularly useful in access layer switches where end devices should never send BPDUs.

Root Guard

The **Root Bridge** plays a central role in STP, determining the best loop-free paths for traffic. If an unauthorized switch with a **lower Bridge ID** (priority + MAC address) is connected to the network, it could inadvertently or maliciously become the Root Bridge, altering the topology and potentially degrading network performance.

Root Guard prevents this by restricting specific ports from becoming Root Ports. When enabled on a port, if a superior BPDU (indicating a better Root Bridge) is received, the port is placed into a **root-inconsistent** state, effectively blocking it from influencing the Root Bridge election. Once the superior BPDUs stop, the port automatically returns to normal operation. Root Guard is typically applied on ports where no upstream Root Bridge should be allowed, such as those connecting to access layer switches.

BPDU Filter

While **BPDU Guard** reacts to unexpected BPDUs, **BPDU Filter** prevents BPDUs from being sent or received on specific ports altogether. This feature can be applied globally or on a per-port basis:

- **Globally:** BPDU Filter allows initial BPDUs to be sent when a port comes up but suppresses further BPDU exchange, making the port behave as if STP is disabled.
- **Per-Port:** When applied directly to a port, BPDU Filter prevents all BPDUs from being sent or received, effectively removing the port from STP participation.

Caution: Improper use of BPDU Filter can lead to **network loops**, as the port is unaware of STP topology changes. It should be used in controlled environments where the risk of loops is minimal.

Loop Guard

Loop Guard is designed to prevent loops caused by **unidirectional link failures** in STP. In normal operation, a port in a **blocking** or **root port** state relies on receiving BPDUs from neighboring switches to maintain its status. If BPDUs unexpectedly stop arriving due to a failure, the port might incorrectly transition to the **forwarding** state, creating a loop.

When **Loop Guard** is enabled, if a port stops receiving BPDUs on a **non-designated port** (i.e., a blocking or root port), the port enters a **loop-inconsistent** state instead of transitioning to forwarding. This proactive measure prevents loops by ensuring ports stay in a safe state until the issue is resolved. Loop Guard is particularly useful on redundant links where unidirectional failures could silently cause loops.

UDLD (Unidirectional Link Detection)

Unidirectional Link Detection (UDLD) is a Layer 2 protocol that detects unidirectional physical link failures, which can lead to network loops or black holes. Such failures occur when traffic flows in one direction but not the other, often due to fiber optic issues, hardware faults, or misconfigurations.

UDLD operates by exchanging **hello packets** between devices on both ends of a link. If one device stops receiving these packets while still sending its own, UDLD detects the inconsistency. Depending on the mode:

- In **normal mode**, UDLD alerts administrators but does not take immediate action.
- In **aggressive mode**, UDLD automatically **disables** the affected port to prevent potential loops.

UDLD is particularly important in fiber optic networks, where physical link issues might not be immediately evident.

Conclusion

Securing the **Spanning Tree Protocol** topology is essential for maintaining a reliable and stable network. Features like **BPDU Guard** and **Root Guard** prevent unauthorized devices from influencing the STP topology, while **BPDU Filter** and **Loop Guard** ensure proper BPDU handling and loop prevention. Additionally, **UDLD** plays a critical role in detecting and mitigating unidirectional link failures. By implementing these protective mechanisms, network administrators can significantly enhance the robustness and resilience of their Layer 2 networks, reducing the risk of outages, loops, and other network disruptions.

Lab Exercises: Protecting the Spanning Tree Protocol Topology

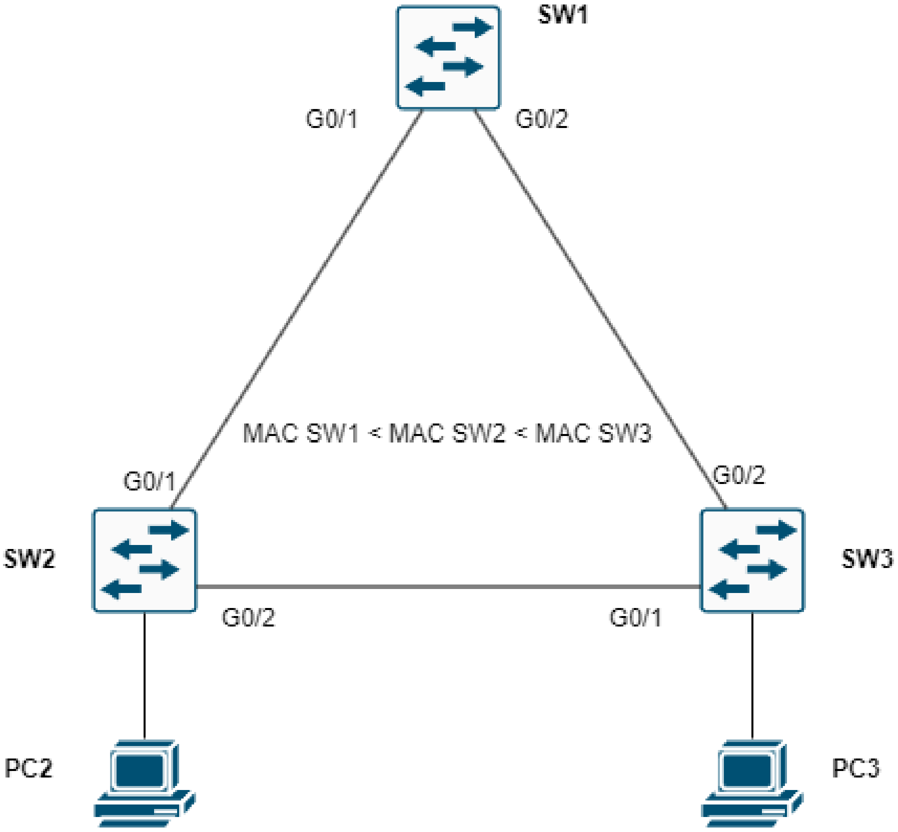

Figure 4.1:

Network Topology Overview:

- **SW1** is connected to **SW2** and **SW3**.
- **SW2** is connected to **PC2** and has a direct link to **SW3**.
- **SW3** is connected to **PC3**.
- **MAC Address Order:**
  MAC SW1 < MAC SW2 < MAC SW3
  (SW1 has the lowest MAC address and will likely become the Root Bridge unless configured otherwise.)

Configuring BPDU Guard

Objective: Protect access ports from unauthorized devices sending BPDUs, which could cause topology changes.

1. **Enable BPDU Guard on Access Ports (Connected to PCs):**
   Since **PC2** and **PC3** are end devices, their interfaces should not receive BPDUs. If a BPDU is received, BPDU Guard will shut down the port to prevent loops.

   On SW2 (Port connected to PC2) and SW3 (Port connected to PC3):

2. **Test the Configuration:**
   Connect a switch (instead of a PC) to **G0/2** on **SW2** and send BPDUs.

   Expected Result: The port should move to an **err-disabled** state. Verify using:

3. **Recover the Port:**

Configuring Root Guard

Objective: Prevent unauthorized switches from becoming the Root Bridge.

1. **Enable Root Guard on Specific Ports:**
   To ensure **SW1** remains the Root Bridge, apply Root Guard on **SW2** and **SW3** ports facing other switches.

   On SW2 (Facing SW3) and SW3 (Facing SW2):

2. **Test the Configuration:**
   Lower the bridge priority on **SW3** to try to make it the Root Bridge:

   Expected Result: The port on **SW2** connected to **SW3** should go into a **root-inconsistent** state, preventing **SW3** from becoming the Root Bridge.

3. **Verify the Root Guard Status:**

Here is some regular text.

Click here to reveal the code!

Exercise 3: Configuring BPDU Filter

Objective: Suppress BPDU transmission on specific ports to prevent unnecessary STP participation.

1. **Enable BPDU Filter on Access Ports (Connected to PCs):**
   On SW2 (Port connected to PC2):

   ```
   interface g0/2
   spanning-tree bpdufilter enable
   ```

   On SW3 (Port connected to PC3):

   ```
   interface g0/1
   spanning-tree bpdufilter enable
   ```

2. **Test the Configuration:**
   Verify that BPDUs are not being sent on these interfaces:

   ```
   debug spanning-tree bpdu send
   ```

3. **Warning:**
   BPDU Filter disables STP on the interface, which can lead to network loops if a switch is accidentally connected.

Exercise 4: Configuring Loop Guard

Objective: Prevent loops caused by unidirectional link failures in the STP topology.

1. **Enable Loop Guard on Non-Designated Ports:**
   Loop Guard should be applied to **root ports** and **alternate ports** that might stop receiving BPDUs.

   On SW2 and SW3 (Ports facing SW1):

   ```
   interface g0/0
   spanning-tree guard loop
   ```

2. **Simulate a Unidirectional Link Failure:**
   Disconnect the **TX (Transmit)** fiber cable from **SW1's G0/1** while keeping **RX (Receive)** connected.

   Expected Result: The port on **SW2** should enter a **loop-inconsistent** state instead of transitioning to forwarding.

3. **Verify Loop Guard Status:**

   ```
   show spanning-tree inconsistentports
   ```

Exercise 5: Configuring UDLD (Unidirectional Link Detection)

Objective: Detect and disable unidirectional links to prevent loops.

1. **Enable UDLD on Fiber Links Between Switches:**
   Apply UDLD on the **G0/1** and **G0/2** interfaces between switches.

   On SW1:

   ```
   interface range g0/1 - 2
   udld aggressive
   ```

   On SW2 and SW3 (Interfaces facing SW1):

   ```
   interface g0/0
   udld aggressive
   ```

2. **Test UDLD Functionality:**
   Disconnect one side of the fiber link (e.g., **TX** on **SW1 G0/1**).

   Expected Result: UDLD will detect the unidirectional link and place the port in an **err-disabled** state.

3. **Verify UDLD Status:**

   ```
   show udld interface
   ```

4. **Recover UDLD-disabled Ports:**

   ```
   interface g0/1
   udld reset
   ```

Discussion Questions:

1. What happens when BPDU Guard is triggered? How is this different from BPDU Filter?
2. How does Root Guard prevent unauthorized topology changes, and in what scenarios would it be most useful?
3. What is the difference between Loop Guard and UDLD in handling unidirectional link failures?
4. After enabling UDLD, how quickly did the network detect and respond to a unidirectional failure compared to standard STP behavior?