

Spanning Tree Protocol (STP) Workbook

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

Mehdi JAFARI ZADEH

Date: February 15, 2025

Contents

1	Spanning Tree Protocol (STP)	5
2	PVST (Per-VLAN Spanning Tree)	13
3	Protecting the Spanning Tree Protocol (STP) Topology	25
4	Rapid Per-VLAN Spanning Tree (Rapid PVST)	33
4.0.1	Network Topology Overview	35
5	Multiple Spanning Tree (MST)	39

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.

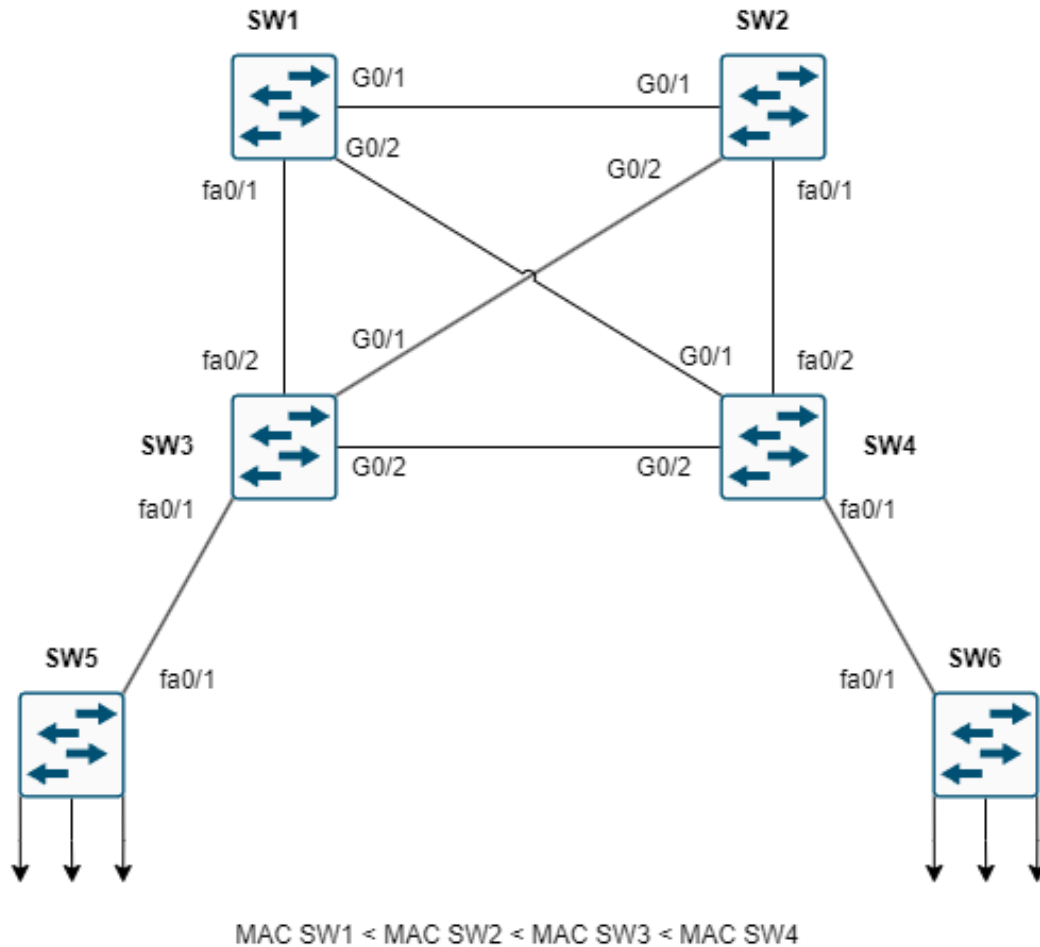


Figure 1.1:

Step-by-Step Tasks

1. Switch STP Mode to Traditional STP

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

[Click here to display the answer:](#)

2. Disable Unused Interfaces

Turn off any extra or unused interfaces to prevent unnecessary loops.

[Click here to display the answer:](#)

3. Set Links to Point-to-Point Mode

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

[Click here to display the answer:](#)

4. Verify Bridge IDs

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

[Click here to display the answer:](#)

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?

[Click here to display the answer:](#)

6. 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?

[Click here to display the answer:](#)

7. How Many Root Bridges Are There in a Network?

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

[Click here to display the answer:](#)

8. How Many Root Ports Are There in a Network?

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

[Click here to display the answer:](#)

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

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

[Click here to display the answer:](#)

10. Estimate the Designated Ports on Each Switch

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

11. Verify Designated Ports Using Commands [Click here to display the answer:](#)**12. 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?

[Click here to display the answer:](#)

13. 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.

[Click here to display the answer:](#)

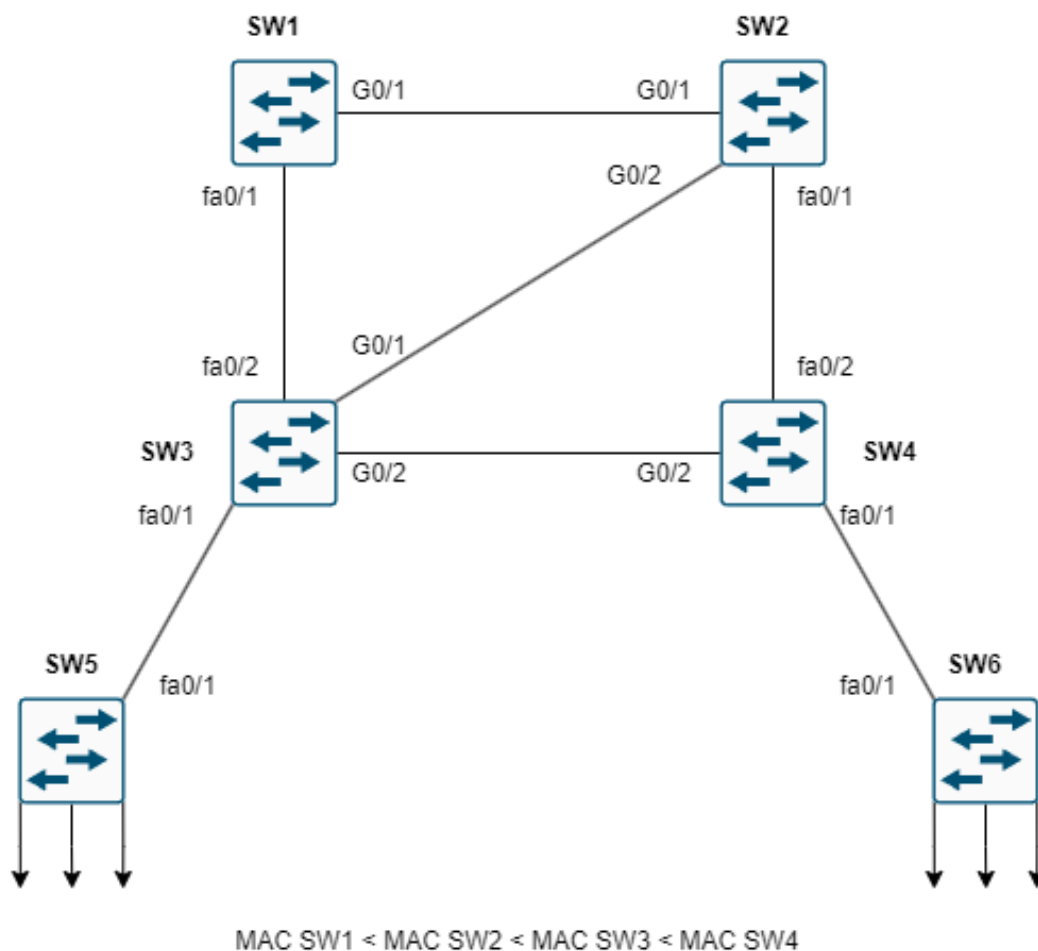


Figure 1.2:

14. 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.
[Click here to display the answer:](#)

Additional Exercises

Draw the Active Topology for Each Scenario

[Click here to display the answer:](#)

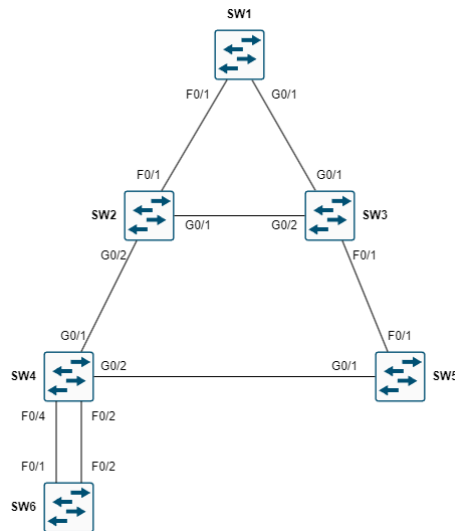


Figure 1.3: *Scenario 1 Topology*

[Click here to display the answer:](#)

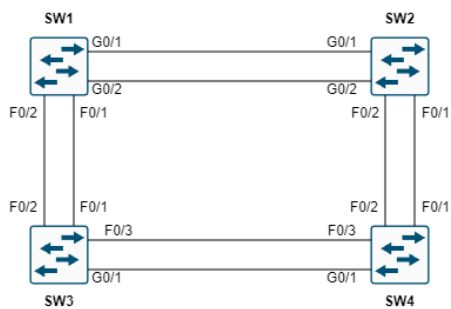


Figure 1.4: *Scenario 2 Topology*

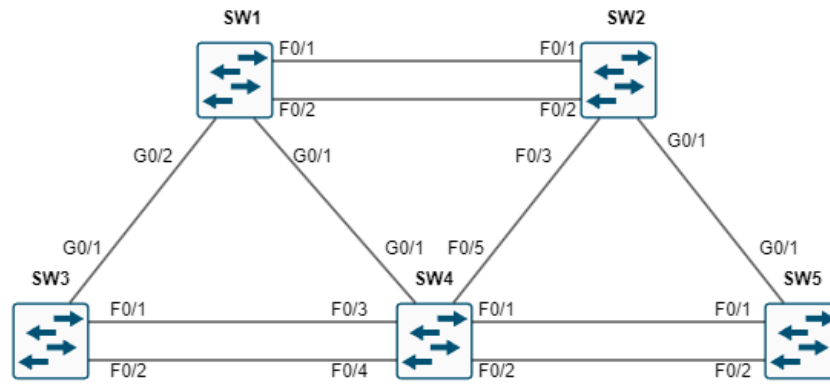


Figure 1.5: *Scenario 3 Topology*

[Click here to display the answer:](#)

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 VLAN Compatibility with Cisco Devices:

PVST 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.

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.

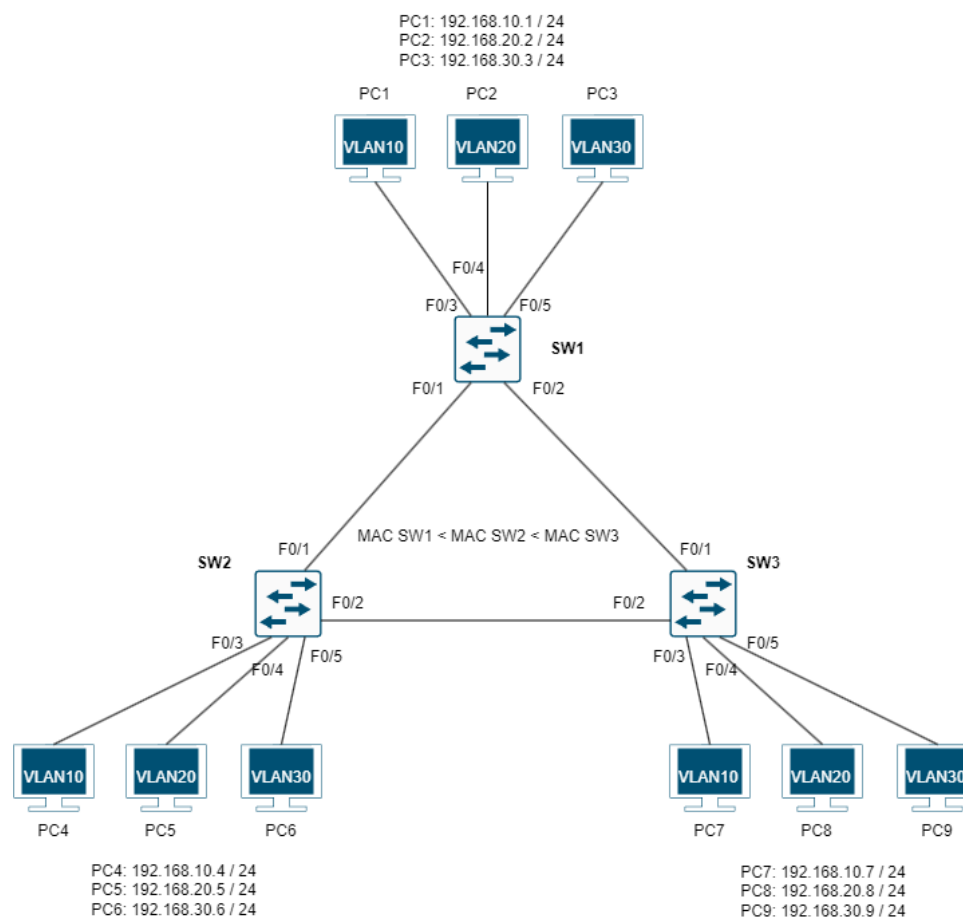


Figure 2.1:

- **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.

[Click here to display the answer:](#)

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.

[Click here to display the answer:](#)

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):

[Click here to display the answer:](#)

- SW2 and SW3 Configuration (VTP Clients):

[Click here to display the answer:](#)

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**.

[Click here to display the answer:](#)

5. Verify STP Status After VLAN Creation

After setting up VTP and VLANs, check how **PVST** has adjusted the spanning tree roles.

[Click here to display the answer:](#)

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):

[Click here to display the answer:](#)

- SW2 (Root for VLAN 20):

[Click here to display the answer:](#)

- SW3 (Root for VLAN 30):

[Click here to display the answer:](#)

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.

[Click here to display the answer:](#)

3. Draw the active topology of each violin.

- SW1 (Root for VLAN 10):
- SW2 (Root for VLAN 20):
- SW3 (Root for VLAN 30):

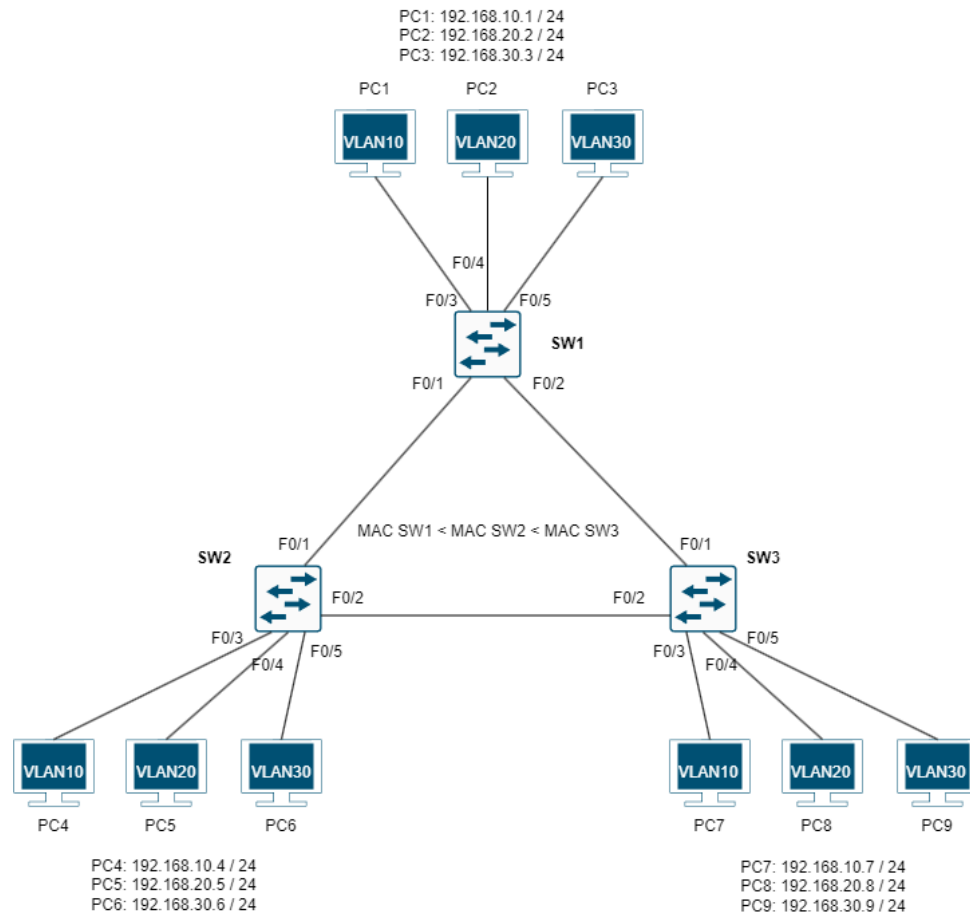


Figure 2.2:

[Click here to display the answer:](#)

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).

[Click here to display the answer:](#)

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**.

[Click here to display the answer:](#)

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**:

[Click here to display the answer:](#)

5. Observe and Estimate New Topology

After shutting down the interfaces:

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

[Click here to display the answer:](#)

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**:

[Click here to display the answer:](#)

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**.

[Click here to display the answer:](#)

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:

[Click here to display the answer:](#)

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**:

[Click here to display the answer:](#)

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**:

[Click here to display the answer:](#)

11. **Verify Uplink Status on SW2**

Task: Check if **UplinkFast** is working correctly:

[Click here to display the answer:](#)

Chapter 3

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

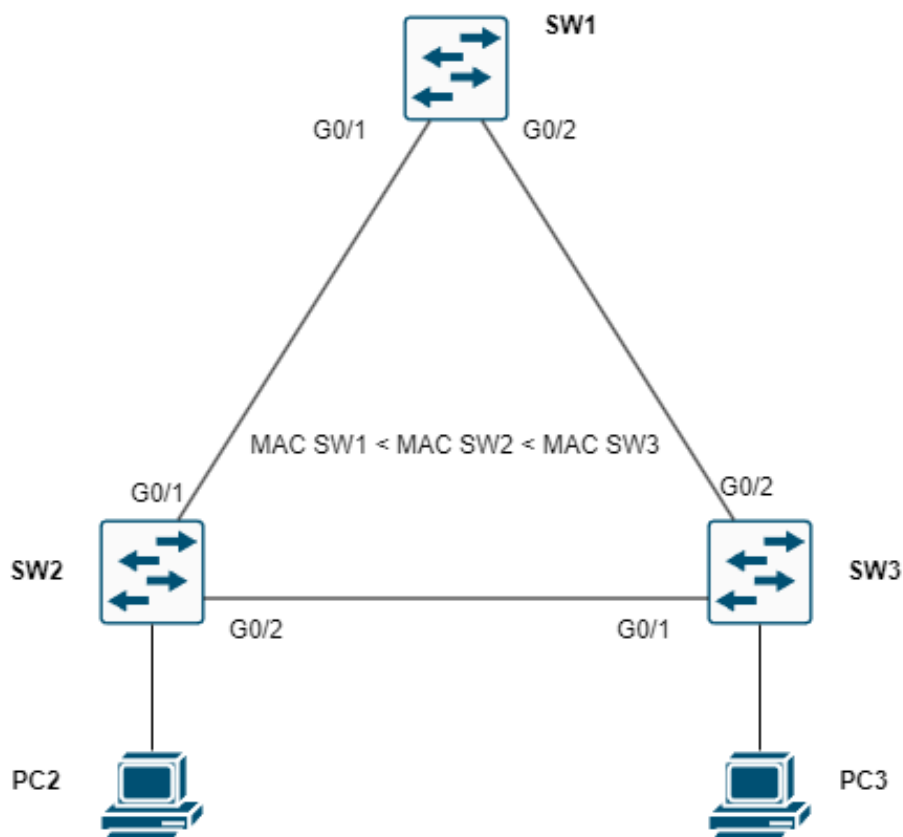


Figure 3.1:

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:**
 $\text{MAC SW1} < \text{MAC SW2} < \text{MAC SW3}$
 (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):

[Click here to display the answer:](#)

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:

[Click here to display the answer:](#)

3. Recover the Port:

[Click here to display the answer:](#)

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):

[Click here to display the answer:](#)

2. Test the Configuration:

Lower the bridge priority on **SW3** to try to make it the Root Bridge:

[Click here to display the answer:](#)

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:

[Click here to display the answer:](#)

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):

[Click here to display the answer:](#)

On SW3 (Port connected to PC3):

[Click here to display the answer:](#)

2. Test the Configuration:

Verify that BPDUs are not being sent on these interfaces:

[Click here to display the answer:](#)

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):

[Click here to display the answer:](#)

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:**

[Click here to display the answer:](#)

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:

[Click here to display the answer:](#)

On SW2 and SW3 (Interfaces facing SW1):

[Click here to display the answer:](#)

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:**

[Click here to display the answer:](#)

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

[Click here to display the answer:](#)

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?

Chapter 4

Rapid Per-VLAN Spanning Tree (Rapid PVST)

Introduction to Rapid PVST

Rapid Per-VLAN Spanning Tree (**Rapid PVST**) is a Cisco proprietary enhancement of the Spanning Tree Protocol (STP) that operates per VLAN while incorporating the rapid convergence benefits of Rapid Spanning Tree Protocol (RSTP). Rapid PVST allows each VLAN to have its own independent spanning tree instance, providing greater flexibility and network optimization in Layer 2 networks.

Key Features of Rapid PVST

- **Per-VLAN Instance:** Unlike standard RSTP, which runs a single instance for the entire network, Rapid PVST maintains a separate spanning tree for each VLAN, allowing better load balancing and path optimization.
- **Rapid Convergence:** By leveraging RSTP (802.1w) mechanisms, Rapid PVST provides faster recovery from topology changes compared to traditional STP (802.1D).
- **Port Roles & States:** Rapid PVST uses the same port roles as RSTP—Root, Designated, Alternate, and Backup—ensuring efficient network operation with minimal downtime.
- **BPDU Exchange:** Bridge Protocol Data Units (BPDUs) are sent out every hello interval (default: 2 seconds) on all ports, enabling rapid detection of topology changes.
- **Backwards Compatibility:** Rapid PVST can interoperate with traditional STP but may fall back to legacy STP behavior if necessary.

Advantages of Rapid PVST

1. **Faster Convergence** – Utilizes rapid transition mechanisms like Proposal/Agreement handshaking for faster network recovery.
2. **Enhanced Load Balancing** – By running a separate instance for each VLAN, traffic can be distributed efficiently across different spanning trees.
3. **Loop Prevention** – Maintains a loop-free topology by dynamically adjusting port states in response to network topology changes.
4. **Optimized Network Performance** – By isolating VLANs, network resources are used efficiently, reducing unnecessary traffic forwarding.

Configuring Rapid PVST on Cisco Switches

To enable Rapid PVST on a Cisco switch, use the following command in global configuration mode:

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

To configure a switch as the root bridge for a specific VLAN:

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

To adjust the bridge priority for better root bridge election control:

```
Switch(config)# spanning-tree vlan <VLAN_ID> priority <value>
```

(Default priority is 32768, lower values increase priority.)

Conclusion

Rapid PVST enhances the traditional Spanning Tree Protocol by providing per-VLAN instances with rapid convergence benefits. It is an essential protocol for networks requiring fast recovery, load balancing, and improved performance across multiple VLANs. Understanding and implementing Rapid PVST effectively ensures a stable and efficient Layer 2 topology in enterprise networks.

Lab Exercises: Rapid Per-VLAN Spanning Tree

4.0.1 Network Topology Overview

The network topology consists of four switches (SW1, SW2, SW3, and SW4) and four personal computers (PC1, PC2, PC3, and PC4). The switches are interconnected in a redundant manner, forming a robust network infrastructure to ensure reliability and fault tolerance.

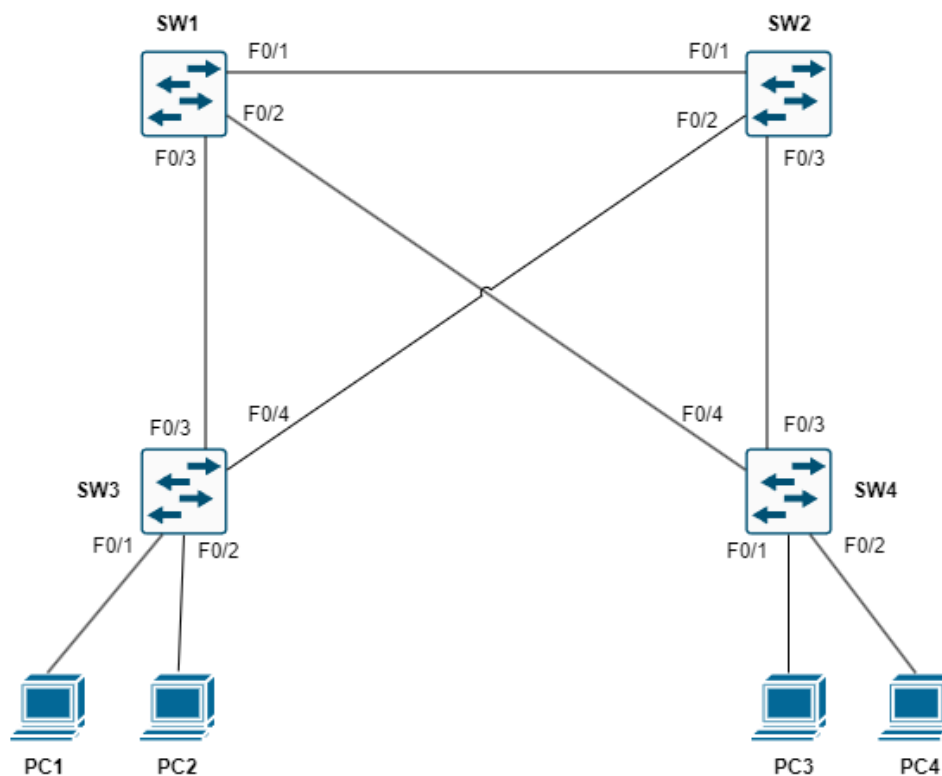


Figure 4.1:

1. Enable RSTP on All Switches

- Configure **RSTP (Rapid PVST+)** on all switches in this topology.
- Verify that RSTP is running correctly.

[Click here to display the answer:](#)

Question: What is the current **Root Bridge**, and how was it selected?

[Click here to display the answer:](#)

2. Manually Set SW1 as the Root Bridge

- Configure **SW1** as the **Root Bridge** for VLAN 1.
- Verify that SW1 is now the Root Bridge.

[Click here to display the answer:](#)

3. Configure Edge Ports for Faster PC Connectivity

- Enable **PortFast** on all PC-connected ports (**SW3 & SW4**).
- Test if these ports enter **forwarding state** immediately after connecting a device.

[Click here to display the answer:](#)

Question: Why is PortFast useful, and why shouldn't it be used on switch-to-switch links?

[Click here to display the answer:](#)

4. Simulate a Link Failure and Observe RSTP Recovery

- **Shutdown the link** between **SW1 and SW2**, then observe how RSTP reacts.
- Which ports change their role to compensate for the failure?

[Click here to display the answer:](#)

Question: How fast did RSTP reconverge compared to classic STP (50s convergence)?

[Click here to display the answer:](#)

5. Observe the Proposal-Agreement Process in RSTP

- Enable **debugging** to watch the RSTP **Proposal-Agreement** process.

[Click here to display the answer:](#)

Question: What happens when a switch detects a new Root Bridge or topology change?

[Click here to display the answer:](#)

Disable Debugging (After Observation):

[Click here to display the answer:](#)

6. Reduce the Network Diameter to Speed Up Convergence

- Lower the STP **diameter** value to **2** to speed up BPDU propagation.

[Click here to display the answer:](#)

Question: How does reducing the network diameter affect STP timers?

[Click here to display the answer:](#)

7. Adjust STP Timers for Faster Convergence

- Reduce the **Forward Delay** and **Max Age** timers to optimize RSTP performance.

[Click here to display the answer:](#)

8. Configure UplinkFast on SW3 and SW4

- Allow **faster failover** when a Root Port fails.

[Click here to display the answer:](#)

Test: Disable the primary Root Port and see how quickly the **backup link** takes over.

Chapter 5

Multiple Spanning Tree (MST)

Introduction to MST

Multiple Spanning Tree (**MST**) is an advanced version of the Spanning Tree Protocol (**STP**) that allows multiple VLANs to be mapped to a single spanning tree instance. MST is standardized under **IEEE 802.1s** and provides a scalable, efficient way to manage Layer 2 network redundancy while optimizing network performance.

Key Features of MST

- **Instance-Based Spanning Trees:** Unlike Per-VLAN Spanning Tree (**PVST+** and **Rapid PVST**), which creates a separate spanning tree instance for each VLAN, MST groups multiple VLANs into a single spanning tree instance (**MSTI**), reducing CPU and memory overhead on network devices.
- **Improved Scalability:** Since fewer spanning tree instances run in the network, MST significantly reduces processing overhead compared to PVST+ or Rapid PVST.
- **Faster Convergence:** MST incorporates the rapid convergence mechanisms of **Rapid Spanning Tree Protocol (RSTP – IEEE 802.1w)**, allowing for quicker network recovery.
- **Loop Prevention:** Ensures a loop-free topology by dynamically managing redundant links within each MST instance.
- **Interoperability with Other STP Versions:** MST can interoperate with traditional STP, RSTP, and PVST+ networks through boundary ports that function as STP/PVST interaction points.

MST Region and Instances

MST introduces the concept of **MST Regions**, which are groups of switches configured with the same MST parameters, including:

1. **Region Name** – A user-defined name that must match across all switches in the region.
2. **Configuration Revision Number** – A manually set number used to track MST configuration changes.
3. **VLAN-to-Instance Mapping** – A table that defines which VLANs belong to which MST instance.

Each MST region supports the following instances:

- **MST Instance 0 (IST - Internal Spanning Tree):** A default instance that interacts with other spanning tree protocols (STP/PVST).
- **MST Instance 1 and Higher:** Additional spanning tree instances that allow VLAN load balancing across different paths.

Advantages of MST

1. **Reduces CPU and Memory Load** – By aggregating VLANs into fewer spanning tree instances, MST lowers resource consumption compared to PVST+.
2. **Efficient VLAN Load Balancing** – By assigning VLANs to different MST instances, network traffic can be better distributed, preventing bottlenecks.
3. **Faster Network Convergence** – Uses RSTP mechanisms for rapid network recovery during topology changes.
4. **Interoperability** – Can coexist with legacy STP/PVST environments through MST boundary ports.
5. **Scalability** – Supports large enterprise networks with thousands of VLANs by reducing the number of spanning tree instances.

Configuring MST on Cisco Switches

To enable MST on a Cisco switch, follow these steps:

1. **Enable MST Mode:**

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

2. **Enter MST Configuration Mode:**

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

3. **Define the Region Name and Revision Number:**

```
Switch(config-mst)# name <region_name>  
Switch(config-mst)# revision <number>
```

4. **Map VLANs to MST Instances:**

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

5. **Exit and Apply Configuration:**

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

6. **Verify MST Configuration:**

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


Conclusion

MST provides a highly scalable and efficient way to manage spanning tree in large Layer 2 networks. By grouping VLANs into fewer spanning tree instances, MST minimizes CPU and memory usage while allowing for efficient traffic load balancing. Its compatibility with RSTP ensures rapid convergence, making it an optimal choice for modern enterprise networks.

Lab Exercises: Multiple Spanning Tree (MST)

This set of questions is designed to reinforce understanding and hands-on skills in **MST (Multiple Spanning Tree Protocol)** and **VTP (VLAN Trunking Protocol) v2**. These questions focus on deeper aspects of configuration, troubleshooting, optimization, and security.

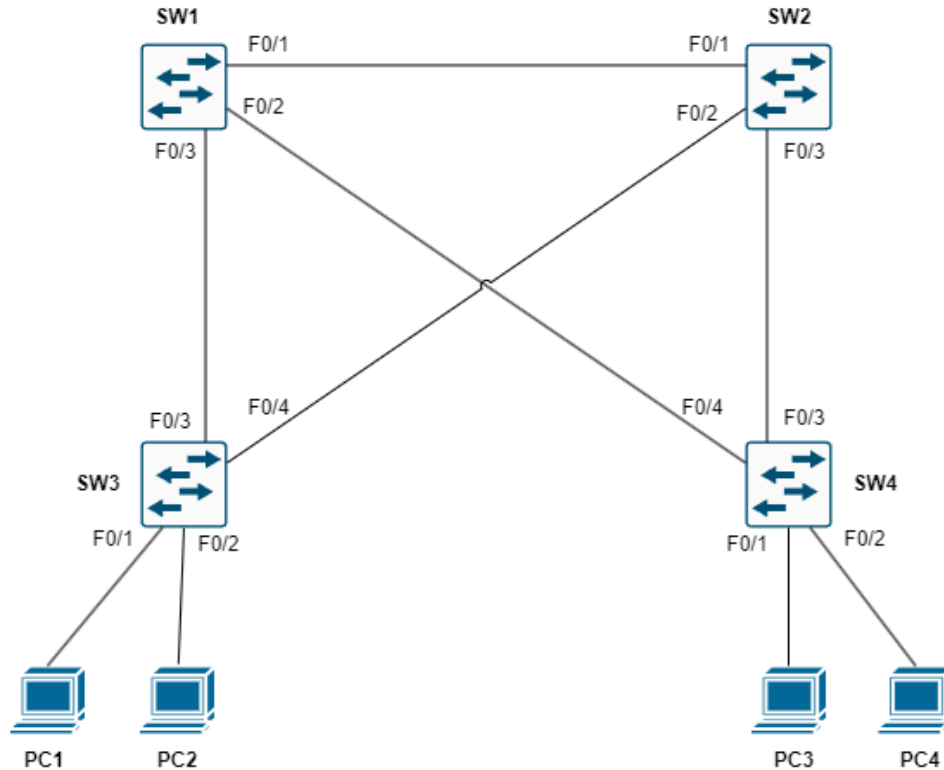


Figure 5.1:

1. **Enable VTP and Set Server Mode on SW1 and VTP Client Mode on SW2, SW3, and SW4**
 - Set the **VTP domain name** to **network** and the **password** to **network**.
 - Verify that all switches are using the same VTP domain and password.

[Click here to display the answer:](#)

2. Verify VTP Configuration

[Click here to display the answer:](#)

Since SW1 is the VTP Server, VLANs must be created here, and they will propagate to SW2, SW3, and SW4.

3. Create VLANs 1-30 on SW1

[Click here to display the answer:](#)

4. Verify VLAN Propagation

[Click here to display the answer:](#)

- Ensure VLANs 1-30 are listed on all switches.

We now configure Multiple Spanning Tree (MST) on all switches.

5. Change the Spanning Tree Mode to MST

[Click here to display the answer:](#)

6. Verify MST Mode is Enabled

7. [Click here to display the answer:](#)

The MST region name must match across all switches for them to share the same MST configuration.

8. Set MST Name and Revision Number on All Switches

[Click here to display the answer:](#)

9. Verify MST Configuration

[Click here to display the answer:](#)

- Ensure MST Name = network and Revision Number = 1 on all switches.

We now assign VLANs to different MST Instances.

10. Configure VLAN-to-Instance Mapping on All Switches

[Click here to display the answer:](#)

11. Verify MST Instance Mapping

[Click here to display the answer:](#)

- VLANs should be correctly mapped to their respective MST instances.

Now, we configure the Root Bridge per MST Instance.

12. Set SW1 as Root Bridge for MST Instance 1

[Click here to display the answer:](#)

13. **Set SW2 as Root Bridge for MST Instance 2**

[Click here to display the answer:](#)

14. **Set SW3 as Root Bridge for MST Instance 3**

[Click here to display the answer:](#)

15. **Verify Root Bridge Assignment**

[Click here to display the answer:](#)

- SW1 should be the Root for MST Instance 1.
- SW2 should be the Root for MST Instance 2.
- SW3 should be the Root for MST Instance 3.

16. **Verify MST Configuration**

[Click here to display the answer:](#)

- Ensure VLAN mappings are correct.
- Verify Root Bridges for each MST Instance.
- Confirm all switches belong to the same MST Region.