Spanning Tree Protocol (STP) Workbook

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

Mehdi JAFARI ZADEH

Date: January 29, 2025

Contents

Chapter 1

Introduction

1.1 Purpose and Audience

Welcome to the **Spanning Tree Protocol (STP) Workbook**. This resource is designed for students and IT professionals who want hands-on practice with the various flavors of STP, such as PVST+, RSTP, and MSTP. By working through the material, you'll build a solid foundation in both the theoretical and practical aspects of Spanning Tree deployment.

- **Who Should Use This Workbook?**
  - Networking learners familiar with **basic Ethernet switching** and **VLAN fundamentals**.
  - Network engineers or administrators looking to strengthen or refresh their STP expertise.
- **What Topics Are Covered?**
  - Core concepts and functions of STP.
  - Configuration steps and relevant commands.
  - Practical lab exercises to reinforce concepts.
  - Advanced discussions of load balancing, security enhancements, and recommended practices.

1.1.1 Prerequisites

Before diving into this workbook, you should have:

- **General CLI experience**: Ability to move between different switch configuration modes (e.g., global config, interface config).
- **IP addressing and subnetting knowledge**: Comfort with IPv4/IPv6 addresses, subnet masks, and interface configurations.
- **Basic VLAN skills**: Understanding VLAN creation, VLAN assignment, and trunk port setup.

1.2 How to Use This Workbook

The workbook is divided into sections that build on each other. Each section features an introduction to theoretical concepts, followed by configuration examples, and concludes with exercises to apply what you've learned.

1. **Theory**: Explains the principles and objectives of each topic.
2. **Command References**: Highlights relevant commands for each concept.

3. **Hands-On Exercises**: Lets you configure and test STP in different scenarios.

4. **Answer Key (at the end)**: Provides detailed solutions and reasoning for the exercises.

Note. *Attempt each exercise on your own first. Use the answer key only as a reference after you've tried to solve the tasks. Experimentation and troubleshooting enhance your understanding of STP and develop valuable problem-solving skills.*

Revisit earlier sections if you find yourself unclear on any points; each new topic often depends on earlier fundamentals. By following this structure, you'll deepen your theoretical knowledge while also learning practical configuration and troubleshooting skills for real-world STP deployments.

Chapter 2

STP Fundamentals

2.1 Redundancy and the Need for STP

2.1.1 Redundancy in Switched Networks

Enterprise networks often implement **redundant connections** between switches to boost reliability and uptime. While extra links improve resilience, they also risk creating **Layer 2 loops**.

What Are Layer 2 Loops?

A Layer 2 loop occurs when multiple paths exist between the same devices and broadcast or certain multicast/unicast frames end up circulating indefinitely. This leads to:

- **Broadcast Storms**: Excessive broadcast traffic overwhelms links and switch CPUs.
- **MAC Table Instability**: Switches repeatedly update their MAC address tables with conflicting data, causing incorrect forwarding.
- **High CPU Usage**: Switches become inundated as they struggle to handle the repeated frames and table updates.

2.1.2 How STP Breaks Loops

The **Spanning Tree Protocol** (as defined in IEEE 802.1D, and later improved in additional standards) systematically identifies and **disables certain ports** in a redundant topology. This ensures there is only one logical path to any segment, forming a "tree" without loops. If a primary path fails, STP rapidly recalculates the topology and activates a previously blocked port to restore connectivity.

2.2 STP Concepts and Terminology

2.2.1 Bridge ID and Root Bridge

- **Bridge ID (BID)**: Unique to each switch, consisting of a **bridge priority** and a **MAC address**.
- **Root Bridge**: The switch with the **lowest BID** acts as the logical hub for the network's spanning tree. All other switches calculate their best path to this root.

2.2.2 Port Roles

- **Root Port (RP)**: On non-root switches, this port leads **directly** to the Root Bridge via the path with the lowest cost.

- **Designated Port (DP)**: One per segment, it forwards traffic away from the Root Bridge. Determined by the lowest path cost and then by BID if there's a tie.
- **Non-Designated/Blocked Port**: Any port not chosen as a DP or RP remains blocked to prevent loops in the network.

2.2.3 Port States (802.1D STP)

1. **Blocking**: Receives BPDUs but does not forward frames.
2. **Listening**: Prepares to participate in forwarding; listens for BPDUs but discards data frames.
3. **Learning**: Learns MAC addresses but still does not forward data frames.
4. **Forwarding**: Actively sends and receives data and BPDUs.
5. **Disabled**: Port is shut down or otherwise administratively disabled.

Note. *Rapid STP (RSTP) and MSTP simplify these states to Discarding, Learning, and Forwarding.*

2.2.4 BPDU (Bridge Protocol Data Unit)

BPDUs carry critical STP information (Bridge ID, root path cost, timers, etc.) between switches. They enable the election of the Root Bridge, selection of port roles, and detection of topology changes.

2.2.5 Path Cost

- **Path Cost** indicates link desirability. Common defaults: 10 Mbps = 100, 100 Mbps = 19, 1 Gbps = 4, 10 Gbps = 2.
- Lower cost implies a more preferred path.

2.3 STP Timers

Three key timers govern the speed and reliability of STP convergence:

1. **Hello Time (default 2 seconds)**
   - How often the Root Bridge sends out BPDUs.
2. **Forward Delay (default 15 seconds)**
   - Duration the port remains in Listening and Learning states before transitioning to Forwarding.
3. **Max Age (default 20 seconds)**
   - Time a switch will retain a received BPDU before considering it invalid.

Note. *Tweaking timers can accelerate or slow down convergence. Misconfiguration can lead to instability.*

2.4 Root Bridge Election

When STP launches, all switches initially assume they can be the root and send out BPDUs with their own Bridge ID. Through BPDU exchanges:

1. Switches compare **Bridge IDs**.
2. The lowest Bridge ID wins the **Root Bridge** title.

3. Non-root switches compute their route cost to the Root.
4. Each switch designates one **Root Port** with the smallest cost path to the Root.
5. Every segment has one **Designated Port** (lowest cost/BID).
6. Other ports become **blocked** to avoid loops.

2.4.1 Tie-Breakers

If two switches have the same priority, the switch with the lower **MAC address** wins. For a given switch's Root Port selection, if path costs are equal, the decision continues with comparing sending BIDs or port IDs.

2.5 Basic STP Configuration

Below are examples using a Cisco CLI, though specifics may differ by device model or vendor.

2.5.1 Setting the STP Mode

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

Note. ***PVST+*** *is Cisco's Per-VLAN Spanning Tree enhancement, but classic STP and PVST+ share similar fundamental mechanics.*

2.5.2 Configuring the Root Bridge

```
Switch(config)# spanning-tree vlan 10 priority 4096
```

Lower priority = higher likelihood of becoming Root. Multiples of 4096 are commonly used.
Alternatively:

```
Switch(config)# spanning-tree vlan 10 root primary
```

This automatically adjusts the priority so that your switch takes over as root for VLAN 10.

2.5.3 PortFast on Edge Ports

PortFast allows access ports (toward end-user devices) to bypass Listening/Learning:

```
Switch(config-if)# spanning-tree portfast
```

Caution: Only enable on ports connected to **end devices**, never inter-switch trunk ports.

2.5.4 Verification

Use these commands to check STP status:

```
Switch# show spanning-tree
Switch# show spanning-tree vlan 10
Switch# show spanning-tree detail
```

These commands provide details on the Root Bridge, port roles, costs, and configured timers.

2.6 Exercises

Exercise 2.1: Observing a Basic Topology

- **Objective**: Watch STP elect a root automatically.
- **Setup**:
  - Connect three switches (A, B, and C) in a triangle.
  - Verify STP is active.
- **Tasks**:
  1. Power up and let STP converge (30–60 seconds).
  2. Use `show spanning-tree` on each switch to find:
     - The Root Bridge.
     - Root Ports vs. Designated Ports.
     - Any blocked ports.
  3. Review each switch's **Bridge ID**.
- **Challenge:**
  - Why was the winning Root Bridge selected?
  - If a tie occurred, what resolved it?

Exercise 2.2: Forcing a Root Bridge

- **Objective**: Manually set which switch becomes root.
- **Setup**: Same three-switch triangle.
- **Tasks**:
  1. On **Switch B**, run:

     ```
     spanning-tree vlan 10 priority 4096
     ```

  2. Verify via `show spanning-tree vlan 10` that Switch B is now the root.
  3. Check port role changes.
- **Challenge:**
  - How would you make Switch A root for VLAN 20?
  - Why is it advantageous to assign different root switches for different VLANs?

Exercise 2.3: Using PortFast and BPDU Guard

- **Objective**: Speed up edge port convergence and protect against loops.
- **Setup**: Switch A with two access ports connected to two PCs (PC1, PC2).
- **Tasks**:
  1. Enable PortFast on the PC-connected interfaces.
  2. Enable BPDU Guard on those interfaces:

```
interface range FastEthernet0/1 - 2
  spanning-tree portfast
  spanning-tree bpduguard enable
```

3. Unplug and reconnect the PCs; note the instant Forwarding state.

- **Challenge:**
  - What happens if you link another switch to a PortFast + BPDU Guard port?

Exercise 2.4: Adjusting Path Costs

- **Objective**: Change STP paths by modifying interface cost.
- **Setup**: Reuse the three-switch triangle.
- **Tasks**:
  1. Identify the current Root Bridge.
  2. On a non-root switch, increase or decrease the cost on one trunk interface:

```
interface <port>
  spanning-tree cost <value>
```

  3. Verify via `show spanning-tree` that a different Root Port was selected.

- **Challenge:**
  - Why might changing interface cost be preferable over adjusting switch priorities in certain designs?

Summary and Next Steps

At this stage, you should be comfortable with the **basic principles of STP**—from why we need it to how it operates and is configured. The next section explores **STP variants (PVST+, RSTP, MSTP)**, diving into improvements in convergence speed, scalability, and flexibility.

Note. *Don't rush forward until you've mastered the fundamentals. Repetition and experimentation in a lab environment are key to truly understanding STP's behavior.*

Chapter 3

STP Variants (PVST+, RSTP, MSTP)

Several enhancements have been introduced to address classic STP limitations, focusing on faster convergence and improved scalability. Here we cover the primary variants: **PVST+**, **RSTP (802.1w)**, and **MSTP (802.1s)**.

3.1 Per-VLAN Spanning Tree Plus (PVST+)

3.1.1 Overview

- **PVST+** is a Cisco-proprietary protocol that maintains a separate spanning tree instance **for each VLAN**.
- It allows **per-VLAN load balancing** by choosing distinct root switches for different VLANs.
- Often the default on many Cisco Catalyst switches.

3.1.2 Differences from 802.1D

- **Per-VLAN**: Instead of a single tree for all VLANs, each VLAN has its own.
- **Fine-Tuned Load Balancing**: Administrators can select different root bridges for each VLAN.

3.1.3 Configuration

1. **Enabling PVST+**

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

   (Often already enabled by default.)

2. **Setting VLAN-Specific Roots**

   ```
   Switch(config)# spanning-tree vlan 10 root primary
   ```

   Automatically adjusts priority to ensure the switch becomes root for VLAN 10.

3. **Load Balancing Example**

   - Make SwitchA root for VLAN 10:

     ```
     SwitchA(config)# spanning-tree vlan 10 root primary
     ```

   - Make SwitchB root for VLAN 20:

     ```
     SwitchB(config)# spanning-tree vlan 20 root primary
     ```

3.1.4 Exercises

- **Exercise 3.1a**: Convert from classic STP to PVST+.
- **Exercise 3.1b**: Designate Switch1 as root for VLAN 10 and Switch2 as root for VLAN 20; confirm roles and port states with `show spanning-tree vlan <id>`.

3.2 Rapid Spanning Tree Protocol (RSTP / 802.1w)

3.2.1 Overview

- **RSTP** improves convergence time significantly over 802.1D.
- It redefines port roles and introduces faster transition to forwarding when the topology changes.

3.2.2 Key Upgrades

1. **New Port Roles**
   - **Alternate Port**: Provides a backup path to the root if the current RP fails.
   - **Backup Port**: Backup for a DP on shared media segments.
2. **Faster Convergence**
   - **Proposal/Agreement** handshake streamlines bringing ports up.
   - States are simplified to **Discarding**, **Learning**, **Forwarding**.
3. **Edge Ports**
   - Similar to PortFast, edge ports move straight to Forwarding if connected to end devices.

3.2.3 Configuration

1. **Enable Rapid PVST+ (Cisco)**

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

2. **Verification**

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

   Look for an indication of `RSTP` or `rapid-pvst`.

3.2.4 Exercises

- **Exercise 3.2a**: Convert to Rapid PVST+ on all switches, then measure link-up to forwarding time.
- **Exercise 3.2b**: Identify **Alternate Ports** in a redundant topology using `show spanning-tree` and observe quick failover.

3.3 Multiple Spanning Tree Protocol (MSTP / 802.1s)

3.3.1 Overview

- **MSTP** groups multiple VLANs into a **small number of MST instances**, reducing overhead compared to having a separate instance for every VLAN.
- Maintains **compatibility** with older STP variants by appropriate boundary interactions.

3.3.2 Concepts

1. **MST Region**: Defined by a shared **Region Name**, **Revision Number**, and identical **VLAN-to-instance** mappings.
2. **MST Instances**: Each instance (MSTI) runs its own spanning tree; VLANs assigned to an instance share that tree.
3. **Internal Spanning Tree (IST)** or **Instance 0**: Manages traffic for any VLANs not explicitly assigned to another instance and interacts with external STP domains.

3.3.3 Configuration (Cisco Example)

1. **Enable MST**

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

2. **Enter MST Configuration Mode**

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# name MyRegion
Switch(config-mst)# revision 2
```

3. **VLAN Mapping**

```
Switch(config-mst)# instance 1 vlan 10,20
Switch(config-mst)# instance 2 vlan 30,40
```

4. **Apply and Verify**

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

5. **Set Root**

```
Switch(config)# spanning-tree mst 1 root primary
Switch(config)# spanning-tree mst 2 root secondary
```

3.3.4 Exercises

- **Exercise 3.3a**: Configure MST region for three switches, naming the region and assigning VLANs 10/20 to MSTI 1, and VLANs 30/40 to MSTI 2.
- **Exercise 3.3b**: Make SwitchA the primary root for MSTI 1 and SwitchB the primary root for MSTI 2; verify with `show spanning-tree mst <instance>`.
- **Exercise 3.3c**: Alter one switch's region name or revision and observe the mismatch. Then fix the settings to restore a single region.

3.4 Comparison of STP Variants

Feature	PVST+	RSTP (802.1w)	MSTP (802.1s)
Instances	Per VLAN	Per VLAN with Rapid PVST+, or single in pure RSTP	Can create multiple MST instances, each mapped to specific VLANs
Convergence	Faster than 802.1D, but not as quick as RSTP	Very fast (seconds)	Utilizes RSTP mechanics within each instance
Scalability	Can be high overhead with many VLANs	Straightforward in smaller setups	Highly scalable by grouping VLANs into fewer instances
Vendor Support	Cisco proprietary (PVST+)	IEEE standard (802.1w)	IEEE standard (802.1s)
Load Balancing	Per-VLAN root configuration	If using Rapid PVST+, per-VLAN is possible	Assign different VLANs to separate MST instances

Table 3.1: Comparison of STP Variants

3.5 Summary and Next Steps

We've explored **PVST+**, **RSTP**, and **MSTP**. Each has its strengths:

- **PVST+** offers simple per-VLAN load balancing but becomes unwieldy in large environments.
- **RSTP** speeds up convergence significantly.
- **MSTP** scales more elegantly by combining multiple VLANs into fewer spanning tree instances.

In the next section, we'll delve into **advanced STP tuning**—covering PortFast, BPDU Guard, and other enhancements that enhance stability and security.

Chapter 4

Advanced STP Tuning and Security

Having covered the basics and variants of STP, we now look at **advanced features and security mechanisms**. These tools help fine-tune STP for performance, minimize risk from misconfiguration, and defend against malicious or accidental loops.

4.1 STP Features and Enhancements

4.1.1 PortFast

- **Definition**: Immediately transitions an interface to **Forwarding**, skipping Listening and Learning.
- **Use Case**: Ideal on **access ports** connecting to PCs or other endpoints.
- **Warning**: Do not enable on inter-switch or trunk links.

Cisco CLI Example:

```
Switch(config)# interface range Fa0/1-2
Switch(config-if-range)# spanning-tree portfast
```

4.1.2 BPDU Guard

- **Definition**: Shuts down a PortFast-enabled port if a BPDU is received, preventing a rogue switch from influencing STP.
- **Purpose**: Stops loops or root role changes triggered by unapproved devices.
- **Behavior**: Moves the port into **err-disabled** upon detecting a BPDU.

Cisco CLI Example:

```
Switch(config-if)# spanning-tree bpduguard enable
```

Or globally:

```
Switch(config)# spanning-tree portfast bpduguard default
```

4.1.3 Root Guard

- **Definition**: Prevents a port from becoming a root port if a superior BPDU arrives.
- **Purpose**: Keeps the designated Root Bridge intact in the network, stopping unexpected root changes.

- **Behavior**: A port receiving a superior BPDU goes into **root-inconsistent** (blocking) rather than allowing a new root.

Cisco CLI Example:

```
Switch(config-if)# spanning-tree guard root
```

4.1.4 Loop Guard

- **Definition**: Ensures a blocking port remains blocked if BPDUs are lost (unidirectional link).
- **Purpose**: Avoid loops that could form when a port incorrectly transitions to forwarding after it stops seeing BPDUs.
- **Behavior**: Port enters **loop-inconsistent** state if BPDU reception ceases unexpectedly.

Cisco CLI Example:

```
Switch(config-if)# spanning-tree guard loop
```

4.1.5 UDLD (Unidirectional Link Detection)

- **Definition**: Not strictly an STP feature but often used in tandem. Detects fiber or cable faults that create one-way traffic.
- **Behavior**: In **aggressive** mode, UDLD disables a port if neighbor messages are not acknowledged correctly, preventing a hidden loop.
- **Modes**: Normal (logs) vs. Aggressive (disables port).

Cisco CLI Example:

```
Switch(config)# udld aggressive
Switch(config)# interface gi0/1
Switch(config-if)# udld port aggressive
```

4.2 Load Balancing Techniques

1. **Adjusting VLAN Priorities**: Assign different VLANs to different root bridges (e.g., Switch A is root for VLAN 10, Switch B is root for VLAN 20).
2. **Tuning Path Costs**: Alter interface costs to steer traffic on specific links.
3. **Using Multiple Instances (MST)**: Group VLANs to different MST instances to distribute load.

4.3 Configuration Best Practices

- **PortFast + BPDU Guard for Edge Ports**: Quickly bring up access ports and mitigate loops from rogue switches.
- **Root Guard at Distribution/Core**: Keep the intended root stable in multi-layer designs.
- **Loop Guard / UDLD on Trunks**: Protect against unidirectional failures on critical uplinks.
- **Plan Load Balancing**: Distribute VLANs or adjust costs to spread traffic across redundant paths.
- **Consistent STP Mode**: All switches should run the same STP variant to avoid conflicts.

4.4 Exercises

Exercise 4.1: Securing Edge Ports with BPDU Guard

- **Objective**: Prevent loops caused by a rogue switch on an access port.
- **Setup**: Single switch with multiple access ports.
- **Tasks**:
  1. Enable PortFast and BPDU Guard on access ports.
  2. Attach another switch; watch it go err-disabled on receiving a BPDU.
  3. Recover the port by manually shutting and enabling it.
- **Challenge**: What if you enable `spanning-tree portfast bpduguard default` globally?

Exercise 4.2: Root Guard Usage

- **Objective**: Prevent an access layer switch from becoming root in a distribution/core design.
- **Setup**: Core switch and two access switches.
- **Tasks**:
  1. Enable Root Guard on the core-facing ports.
  2. Temporarily set a lower priority on an access switch. Root Guard should block this attempt.
  3. Check `show spanning-tree inconsistentports`.
- **Challenge**: Under what conditions might Root Guard complicate multi-vendor interoperability?

Exercise 4.3: Loop Guard to Prevent Unidirectional Loops

- **Objective**: Block a port if BPDUs suddenly go missing.
- **Setup**: Two or more switches connected by trunks.
- **Tasks**:
  1. Enable Loop Guard on relevant trunk interfaces.
  2. Simulate a unidirectional scenario.
  3. Observe the port going into **loop-inconsistent** state rather than forwarding.
- **Challenge**: Compare Loop Guard vs. UDLD for unidirectional link protection.

Exercise 4.4: Balancing Traffic with MST or VLAN Priorities

- **Objective**: Use MST instances or PVST+ priority to distribute traffic.
- **Setup**: Three-switch triangle, VLANs 10, 20, 30, 40.
- **Tasks**:
  1. **MST approach**: Assign VLANs 10/20 to MSTI 1, VLANs 30/40 to MSTI 2, then pick different root switches.
  2. **PVST approach**: Pick different root switches for each VLAN.
  3. Verify port roles with `show spanning-tree` or `show spanning-tree mst`.
- **Challenge**: Which method is more scalable and why?

4.5 Summary

Advanced STP features like **BPDU Guard**, **Root Guard**, **Loop Guard**, and **UDLD** help protect against loops and minimize disruptions. **Load balancing** can be achieved through careful management of priorities or MST instances. Implementing these best practices fortifies your network against common pitfalls, malicious attacks, and physical link issues.

The following section discusses **Monitoring and Troubleshooting** STP. We'll look at the key commands to view the topology, how to interpret debug outputs, and how to systematically approach common STP issues.

Chapter 5

Monitoring and Troubleshooting

5.1 Key Show Commands

5.1.1 Global STP State

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

- Displays overall STP data for all VLANs or instances.
- Identifies the **protocol** in use (STP, RSTP, MST, etc.), root info, and port roles.

```
Switch# show spanning-tree summary
```

- Shows a summarized STP configuration, including global settings (BPDU Guard, Loop Guard, etc.).

```
Switch# show spanning-tree detail
```

- Comprehensive listing for each VLAN or instance, including BPDU counts, timers, and topology changes.

5.1.2 VLAN or Instance Specific

```
Switch# show spanning-tree vlan <vlan-id>
```

- Shows spanning tree data for a particular VLAN (PVST+/Rapid PVST+).

```
Switch# show spanning-tree mst <instance-id>
```

- For MSTP, displays details for a specific instance (root switch, port roles, etc.).

5.1.3 Interface and Neighbor Info

```
Switch# show interfaces trunk
```

- Shows trunk ports and the VLANs allowed on them.

```
Switch# show interfaces <interface> switchport
```

- Displays access/trunk mode settings and VLAN membership.

```
Switch# show spanning-tree interface <interface> detail
```

- Provides port-specific STP parameters like cost, priority, designated bridge, etc.

5.2 Debug Commands

Use debug commands with care in live environments due to potential CPU load.

```
Switch# debug spanning-tree events
```

- Logs STP changes in real-time (port transitions, topology change notifications).

```
Switch# terminal monitor
```

- Allows debug messages to appear in terminal sessions (Telnet/SSH).

5.3 Common STP Problems

1. **Priority Mismatches**
   - Overlapping priority settings can cause frequent root elections.
2. **Edge Port Misconfiguration**
   - Failure to enable BPDU Guard on user ports risks loops from unauthorized switches.
3. **Unidirectional Links**
   - Missing BPDUs can cause a blocking port to forward unexpectedly. Use **UDLD** or **Loop Guard**.
4. **MST Region Inconsistency**
   - Different name, revision, or VLAN mapping leads to multiple MST regions.
5. **VLAN Pruning Issues**
   - If a VLAN is not allowed on a trunk, its STP process might become isolated.
6. **Physical Layer Defects**
   - Faulty cables or transceivers can trigger loops or flapping ports.

5.4 Troubleshooting Labs

Lab 5.1: Identifying a Loop

- **Scenario**: High CPU, excessive broadcasts, and random MAC flapping.
- **Tasks**:
  1. Check CPU usage, run `show spanning-tree` to find a port in Forwarding that shouldn't be.
  2. Correct the port's mode or enable BPDU Guard.
  3. Confirm network stabilizes.

Lab 5.2: Mystery Root Bridge

- **Scenario**: Intended root for VLAN 10 is Switch A, but Switch C has a lower priority.
- **Tasks**:
  1. `show spanning-tree vlan 10` on each switch.
  2. Adjust priorities to restore Switch A as root.
  3. Verify results with `show spanning-tree vlan 10`.

Lab 5.3: MST Region Mismatch

- **Scenario**: Three MST switches, but one has a different revision.
- **Tasks**:
  1. Use `show spanning-tree mst configuration` to locate mismatches.
  2. Align the region name, revision, and VLAN mappings.
  3. Confirm a single region forms.

Lab 5.4: Debugging PortFast Issues

- **Scenario**: PortFast is enabled on a port connected to a test switch.
- **Tasks**:
  1. `debug spanning-tree events`, connect the test switch.
  2. Observe the BPDU Guard err-disable.
  3. Restore the port after disconnecting the test switch.

5.5 General Troubleshooting Strategy

1. **Gather Symptoms**: Check logs, CPU, and interface counters.
2. **Layer-by-Layer Checks**: Confirm physical and VLAN configurations, then STP details (root, costs, priorities).
3. **Identify the Root Bridge**: Compare expected vs. actual.
4. **Examine Port States**: Look for misconfigured or unexpected Forwarding/Blocking ports.
5. **Check Security Features**: Ensure Root Guard, Loop Guard, BPDU Guard, and UDLD are configured where intended.
6. **Incremental Changes**: Fix one issue at a time, verify results before moving on.

5.6 Summary

Monitoring and troubleshooting STP hinges on mastering key commands to view the network's topology, port roles, and event logs. Knowing typical pitfalls—such as priority misconfigurations, MST region mismatches, or unidirectional links—helps you quickly diagnose and rectify network instability.

Next, we'll bring everything together in **Practical Lab Scenarios**, combining the lessons from all previous sections for a comprehensive exercise in configuring, optimizing, and troubleshooting STP.

Chapter 6

Practical Lab Scenarios

6.1 Scenario 1: Single-VLAN STP

6.1.1 Objective

- Grasp **basic STP** operation in a single VLAN.
- Observe **root election** and port role assignments.

6.1.2 Topology

- **Three switches** in a triangle (A, B, C).
- **VLAN 1** on all trunk interfaces or a single VLAN of choice.

6.1.3 Tasks

1. **Initial Setup**: Clear configs, ensure STP is enabled.
2. **Auto-Election**: Power on, wait for convergence. Use `show spanning-tree` on each switch to see the Root Bridge.
3. **Forcing a Root**: Configure Switch A's priority:

```
SwitchA(config)# spanning-tree vlan 1 priority 4096
```

4. **Verification**: Disconnect/reconnect cables to witness STP re-convergence.

6.1.4 Reflection

- Which switch was initially root?
- How did the manual priority setting affect the topology?

6.2 Scenario 2: Multi-VLAN with PVST+

6.2.1 Objective

- Implement **Per-VLAN Spanning Tree**.
- Use **load balancing** by choosing different roots for different VLANs.

6.2.2 Topology

- **Three or four switches** in a partial mesh.
- VLANs **10** and **20** exist everywhere.

6.2.3 Tasks

1. **Create VLANs**: On each switch, add VLAN 10 and 20, allow them on trunks.
2. **Root Assignments**:
   (a) Switch A is root for VLAN 10:
   ```
   SwitchA(config)# spanning-tree vlan 10 root primary
   ```
   (b) Switch B is root for VLAN 20:
   ```
   SwitchB(config)# spanning-tree vlan 20 root primary
   ```
3. **Verify**: Use `show spanning-tree vlan 10` and `show spanning-tree vlan 20` to confirm root roles and blocked ports.

6.2.4 Reflection

- How did the network partition different VLAN paths?
- Was load balancing achieved?

6.3 Scenario 3: Rapid PVST+

6.3.1 Objective

- Transition to **Rapid PVST+** for faster convergence.
- Observe reduced recovery times after link changes.

6.3.2 Topology

- **Three switches** in a triangle, with VLANs 10 and 20.

6.3.3 Tasks

1. **Initial**: Document root placements under classic STP/PVST+.
2. **Enable Rapid PVST+**:
   ```
   Switch(config)# spanning-tree mode rapid-pvst
   ```
3. **Test Convergence**: Disconnect a trunk; time how quickly forwarding resumes.

6.3.4 Reflection

- How does convergence compare with classic STP?
- Which ports become **Alternate**?

6.4 Scenario 4: MSTP with Four VLANs

6.4.1 Objective

- Implement **MSTP**, assigning VLANs to different MST instances.
- Ensure consistent MST region settings.

6.4.2 Topology

- **Three or four switches**, VLANs **10, 20, 30, 40**.

6.4.3 Tasks

1. **Enable MST**:

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

2. **Map VLANs**:

```
Switch(config-mst)# instance 1 vlan 10,20
Switch(config-mst)# instance 2 vlan 30,40
```

3. **Root Selection**: Switch A root for MSTI 1; Switch B root for MSTI 2.
4. **Check**: Use `show spanning-tree mst configuration` on each switch.

6.4.4 Reflection

- Does MST simplify or complicate your setup for multiple VLANs?
- Were the MST instances created and recognized correctly?

6.5 Scenario 5: Security Hardening

6.5.1 Objective

- Safeguard the network with features like **BPDU Guard**, **Root Guard**, **Loop Guard**, and **UDLD**.

6.5.2 Topology

- **Core/Distribution/Access** setup with multiple VLANs.

6.5.3 Tasks

1. **BPDU Guard** on access ports:

```
Switch(config-if)# spanning-tree portfast
Switch(config-if)# spanning-tree bpduguard enable
```

2. **Root Guard** on core-facing ports:

```
Switch(config-if)# spanning-tree guard root
```

3. **Loop Guard or UDLD** on trunk links:

```
Switch(config-if)# spanning-tree guard loop
```

or

```
Switch(config-if)# udld port aggressive
```

4. **Testing**: Plug in a rogue switch to an access port, attempt to override the root, simulate a unidirectional failure.

6.5.4 Reflection

- Which of these features is most critical in your environment?
- Did you observe the correct error-disable or inconsistent states?

6.6 General Tips for All Labs

- **Diagram Everything**: Visualize your switch interconnections and VLAN design.
- **Compare Before/After**: Use `show spanning-tree` outputs to see changes after each step.
- **Small Steps**: Change one parameter at a time, confirm, then proceed.
- **Troubleshoot Methodically**: If a loop or unexpected behavior appears, revert to a known good state and make incremental adjustments.

6.7 Conclusion

By tackling these **end-to-end lab scenarios**, you consolidate all the knowledge from **basic STP** to **advanced tuning**:

- Setting up **classic STP** and verifying root selection.
- Employing **PVST+** or **Rapid PVST+** for multi-VLAN networks with quicker failover.
- Scaling and load balancing using **MSTP**.
- Hardening STP ports with **PortFast, BPDU Guard, Root Guard, Loop Guard,** and **UDLD**.

Real-world proficiency in STP emerges through **repetition, observation, and troubleshooting**. Continue to modify these labs, add more VLANs, or experiment with varied link costs and see how STP adapts in each scenario.

Chapter 7

Additional Resources

Below are recommendations to further enhance your understanding and stay current with evolving standards and techniques related to Spanning Tree.

7.1 Official Documentation

- **Cisco Official Documentation**
  - Cisco STP Technical Resources
  - PVST+ and RPVST+ Guides
  - MST Deployment Guides
- **IEEE Standards**
  - **IEEE 802.1D** (classic STP)
  - **IEEE 802.1w** (RSTP)
  - **IEEE 802.1s** (MSTP)
  - Available through the IEEE Xplore Library.

7.2 Books and Study Guides

- **Cisco Press**
  - *CCNA 200-301 Official Cert Guide* by Wendell Odom (introductory coverage).
  - *CCNP ENCOR 350-401 Official Cert Guide* by Brad Edgeworth (advanced STP).
  - *CCIE R&S / Enterprise* materials (deep dives into STP tuning).
- *LAN Switching Fundamentals* by Cisco Press – In-depth coverage of Layer 2 switching and STP mechanics.

7.3 Online Articles and Video Tutorials

- **Cisco Learning Network**
  - Community forums discussing STP best practices.
- **NetworkLessons.com**
  - Concise lessons on STP, RSTP, PVST+, MSTP, with lab examples.

- **Popular YouTube Channels**
  - *Network Chuck*, *Keith Barker*, and *David Bombal* often cover STP setups and troubleshooting.

7.4 Lab Simulation Tools

- **Cisco Packet Tracer** – Ideal for CCNA-level STP labs.
- **GNS3 / EVE-NG** – Allows more advanced multi-vendor topologies, suitable for CCNP/CCIE practice.
- **Cisco Modeling Labs (VIRL)** – Official Cisco platform for simulating IOS-based environments.

7.5 Industry Designs and Best Practices

- **Cisco Validated Designs (CVDs)**
  - Blueprint-like documents for enterprise networks, including STP considerations.
  - Cisco Design Zone
- **Enterprise Architecture Guides**
  - Cover campus design, VLAN distribution, and STP hierarchy.

7.6 Conclusion

By leveraging official standards, Cisco documentation, lab tools, and design references, you can further **refine your STP skills**. Continuous practice—configuring, breaking, and fixing STP scenarios—remains the best way to cement your knowledge and become an adept network professional.

Good luck on your journey to mastering Spanning Tree Protocol!