

Multiple Spanning Tree Migration Best Practices

Objective

The objective of this document is to provide you with best practices when migrating to Multiple Spanning Tree (MSTP). Utilizing MSTP over other Spanning Tree variants can improve network efficiency and reliability.

Requirements

- The need to optimize Layer 2 in a mixed hardware environment
 - Cisco Small Business switch(es)
 - Sx250 Series ([Administration Guide](#))
 - Sx300 Series ([Administration Guide](#))
 - Sx350 Series ([Administration Guide](#))
 - SG350X Series ([Administration Guide](#))
 - Sx550X Series ([Administration Guide](#))
 - Cisco Catalyst Switch(es)
- A working understanding of Spanning Tree ([Learn more](#))
- Wireshark (Optional)

Table of Contents

1. [MSTP Terminology](#)
2. [Best Practice #1 - Validate the need for migrating to MSTP](#)
3. [Best Practice #2 - Strategize your migration](#)
4. [Best Practice #3 - Best Practice #3 - Enable point-to-point ports to use PortFast](#)
5. [Best Practice #4 - Enable BPDU Guard on edge ports](#)
6. [Best Practice #5 - Map VLANs to MSTIs, not the IST \(MST0\)](#)
7. [Best Practice #6 - Place all MSTP-enabled switches within the same region](#)
8. [Best Practice #7 - Nest the root bridge of the CIST within the primary MST region](#)
9. [Migration Verification - Is this thing on?](#)
10. [Conclusion](#)

How this guide is structured

This guide will omit steps like logging into the device via SSH, or the management interface - instead we'll highlight the core commands. Each best practice will contain a sub-task outlining appropriate steps for mixed Cisco hardware (Enterprise & SMB). For configuration guides, see the following two links:

- [Configuring MSTP on SMB Switch](#)
- [Configuring MSTP on Catalyst Switch](#)

MSTP Terminology

This section is intended to give you an accessible mental model of the protocol at play. The definitions are interlocking components of the MSTP protocol. Further details are contained in the sub-bullet points.

BPDU - Bridge Protocol Data Unit - These are multicast frames containing all the information a switch needs to continue operating.

Note: that the instance mappings themselves are not in the BPDU.

Region - (*Specific to MSTP*) - A region solves the problem encountered by other STP flavors which sends one BPDU per VLAN. As with Per Vlan Spanning Tree, sending so many BPDUs causes strain on the CPU load and thereby hindering network performance. Instead, with MSTP all VLANs are mapped to a single region.

Instance - An instance is a of logical table of a VLAN, or many VLANs, to a particular region. This instance then maps to a zone. You will complete these steps as a part of your migration.

The default instance 0 (zero), is synonymous with the following terms *MST0*, *Internal Spanning Tree (IST)*.

Any instances that are created by you, are be referenced to as Multiple Spanning Tree Instances, or MSTIs.

This is where good documentation of your network's VLANs will save you headaches.

- If an instance fails, it will not affect other instances.

MSTI - Multiple Spanning Tree Instances - Contains the administratively created instance. These mappings are contained in what's known as an "MRecord", visible via Wireshark. The records encompass details necessary to manage the instance's topology.

IST - Internal Spanning Tree - is the record of switches participating in a MSTP zone. The switches (no matter how many) contained within a zone, are represented to areas outside the zone as a single switch.

- **CST - Common Spanning Tree** - is composed of MSTP regions running its own traditional spanning tree. CST uses links between the switches at the boundary of the MSTP zone.

CIST - Common and Internal Spanning Tree - Composed of both CST and IST that traverses multiple instances based on a shared mapping of VLANs to the instance.

Common and Internal Spanning Tree *is not* Common Spanning Tree.

Now that we've established who this article is for and the pertinent definitions, let's get to the best

practices.

Best Practice #1 - Validate the need for migrating to MSTP

The first best practice involves confirming your need to migrate to MSTP. Understanding your network's existing spanning tree performance is a key factor in this decision. Migrating to MSTP would be a great option for a few reasons, introducing load sharing, which creates the biggest impact to your network efficiency. If layer 2 traffic has increased ahead of your projections, moving to MSTP can increase usefulness/lifetime of your gear via improved performance. Other considerations could be:

Existing STP performance is unsatisfactory - convergence time or the amount of BPDUs transmitted is causing issues

Segment Spanning Tree - reduces the resource load on the switches contained in the MSTP regions.

Mixed hardware environment - MSTP is an open standard, meaning it is great for a mixed vendor environment. It is widely supported.

Note: A common misconception is when migrating to Multiple Spanning Tree you must map one VLAN per instance.

Flavors of spanning tree have sprung up, with variations and twists on previous versions. Compared to Per VLAN Spanning Tree (PVST+), MSTP uses less resources (BPDUS, CPU cycles, transmit time) by maintaining instances of Spanning Tree, or logical versions of Spanning Tree. VLAN traffic is enabled to flow through the layer 2 segments of a network. Forwarding for one port (and VLAN), can also block for a different VLAN. On top of that, if a loop forms in one instance, it will not affect the other instance(s).

Best Practice #2 - Strategize your migration

Once you've validated the need to migrate, ideally, migration is achieved with minimal downtime and existing connectivity is preserved. A little strategy on tackling the migration will go a long way toward ensuring a smooth rollout. To aid with that process, we recommend the following tactical steps.

1. *Document, document, document* - Keeping detailed notes will reduce migration time and potential for errors.

Identify and document all point to point ports, or ports that lead to another switch or router.

Identify and document all edge ports, or ports that lead to an end point like a PC or printer.

Define which VLANs are participating in the migration

Interns are really good at this step!

Determine the order of operations for your network.

Be aware of how a change on one switch can affect a different VLAN.

Schedule downtime for your network, or migrate on the weekend.

Start the migration at the core of your network and work down to distribution and then the access layer.

Best Practice #3 - Enable point-to-point ports to use PortFast

This best practice, and the following one, make good use of all that port documentation. Administrators define an optional parameter on edge ports via the PortFast feature. PortFast prevents Spanning Tree from running on that port. The switch-to-gear facing ports could include a server, workstation, router. The intention is for that port to never bridge the network to another set of open ports. Which could potentially cause loops if the switch received a superior BPDU. Given ports coming online to a network incur a STP calculation on the port, you can save time and CPU load by assigning blocking status ahead of time. It allows the port to rapidly transition to a BPDU sending - forwarding state. Because it has been assigned a status ahead of time.

Note: Ensure ports on the switches are configured for full-duplex transmission.

The steps below will be divided between SMB switches (CLI + GUI) and Enterprise catalyst switches (CLI).

Enabling Portfast on Catalyst Switch - CLI

The CLI commands are presented syntax first, followed by an example of a live command. An extra space has been added after the # to make highlighting for copy > paste a little easier. Text highlighted in blue denotes variables, to be replaced with contextual details from your network. Also note for brevity the only privilege elevation commands we use will be for MSTP Configuration.

```
Catalyst(config)# interface [range(optional)] [port-id]
Catalyst(config-if)# spanning-tree portfast [auto]
```

```
Catalyst(config)# interface range fa0/1 - 24
Catalyst(config-if)# spanning-tree portfast auto
```

Enabling Portfast on SMB Switch - CLI

```
SMBswitch(config)# interface [range(optional)] [port-id]
SMBswitch(config-if)# spanning-tree portfast
```

```
SMBswitch(config)# interface range gi1-15
SMBswitch(config-if)# spanning-tree portfast
```

Enabling Portfast on SMB Switch - GUI

One caveat to note, the SMB switches GUI uses a synonym for *PortFast* - it's known as *Fast Link*.

Step 1. Click **Spanning Tree > STP Interface Settings**.

Step 2. Select an **interface** and click the **Edit** button.

Step 3. Click **Enable** Fast Link.

Note: Remember to apply the changes as well as write the running configuration to the startup configuration.

Best Practice #4 - Enable BPDU Guard on edge ports

This best practice is an extension of the previous one. If a BPDU Guard enabled port sees that the port receiving any superior, topology changing BPDUs, it immediately shuts the port down via *err-disable* status. That would require you to access the switch and resolve the situation.

Note: This may seem like one of those best practices that you might be able to skip. Could you get away with it? Maybe, but for the sake of your future-self, make it so. One errant switch brought onto the network and pumping out erroneous BPDUs, could potentially topple your network.

Enabling BPDU Guard on Catalyst Switch - CLI

```
Catalyst(config)# interface [range(optional)] [port-id]
Catalyst(config-if)# spanning-tree bpduguard enable
```

```
Catalyst(config)# interface range fa0/1 - 24
Catalyst(config-if)# spanning-tree bpduguard enable
```

Enabling BPDU Guard on SMB Switch - CLI

```
SMBswitch(config)# interface [range(optional)] [port-id]
SMBswitch(config-if)# spanning-tree bpduguard enable
```

```
SMBswitch(config)# interface range fa0/1 - 24
SMBswitch(config-if)# spanning-tree bpduguard enable
```

Enabling BPDU Guard on SMB Switch - GUI

Step 1. Log in to the web configuration utility to choose **Spanning Tree > STP Interface Settings**. The STP Interface Settings page opens.

Step 2. Choose the type of **interface** you wish to edit from the Interface Type drop-down list.

Step 3. Click **Go** to show only ports or LAGs on the page.

Step 4. Click the **radio** button of the port or LAG that is connected to the other switch and click **Edit**. The Edit STP Interface window appears.

Step 5. Click the BPDU Guard **Enable** checkbox that corresponds to the desired interface type in the *Interface* field.

Best Practice #5 - Map VLANs to MSTIs, not the IST (MST0)

Now the ports know their appropriate role, let's move on to instance mapping. For best results, limit the amount of instances you create - note there is some nuance. This is counter to best practice and could dissuade an engineer from MSTP as a solution. You may have valid network design considerations for multiple instances, but be aware the best practice is to have a single

instance. Decide what VLANs to map onto the instance(s). Then choose a configuration name and a revision number that will be common to all switches in the network.

Note: When you edit the MSTI VLAN mappings, MSTP restarts.

Mapping VLANs on Catalyst Switch - CLI

```
Catalyst(config)# spanning-tree mst configuration  
Catalyst(config-mst)# instance [instance-id] vlan [vlan-range]
```

```
Catalyst(config)# spanning-tree mst configuration  
Catalyst(config-mst)# instance 1 vlan 1-11
```

Mapping VLANs on SMB Switch - CLI

```
SMBswitch(config)# spanning-tree mst configuration  
SMBswitch(config-mst)# instance [instance-id] vlan [vlan-range]
```

```
SMBswitch(config)# spanning-tree mst configuration  
SMBswitch(config-mst)# instance 1 vlan 1-11
```

Mapping VLANs to MSTI - GUI

Step 1. Click **Spanning Tree > VLAN** to MSTP Instance.

The *VLAN to MSTP Instance* page contains the following fields:

- *MST Instance ID*— All MSTP instances are displayed.
- *VLANs*—All VLANs belonging to the MST instance are displayed.

Step 2. To add a VLAN to an MSTP instance, select the **MST instance**, and click **Edit**.

- *MST Instance ID*—Select the MST instance.
- *VLANs*—Define the VLANs being mapped to this MST instance.
- *Action*—Define whether to Add (map) the VLAN to the MST instance or Remove it.

Step 3. Enter your **parameters**.

Step 4. Click **Apply**. At this point, the MSTP VLAN mappings are established.

Best Practice #6 - Place all MSTP-enabled switches within the same region

The best practice is to place as many switches into a single region as possible. There are no benefits to segmenting the network into multiple regions. As with any routing and switching protocols they require a way to confirm membership to the protocol. The BPDUs sent enable a switch to recognize itself as a member of a particular region. For the bridge to understand their membership of a given region, they must share the following settings:

1. Region name
2. Revision number
3. Digest computed from the VLAN-to-instance mapping

Homing the bridge within a region on Catalyst Switch - CLI

```
Catalyst(config)# spanning-tree mst [instance-id] root primary
```

```
Catalyst(config)# spanning-tree mst 5 root primary
```

Homing the bridge within a region on SMB Switch - CLI

```
SMBswitch(config)# spanning-tree mst configuration  
SMBswitch(config-mst)# instance [instance-id] vlan [vlan-range]  
SMBswitch(config-mst)# name [region-name]  
SMBswitch(config-mst)# revision [revision-id]
```

```
SMBswitch(config)# spanning-tree mst configuration  
SMBswitch(config-mst)# instance 1 vlan 10-20  
SMBswitch(config-mst)# name region1  
SMBswitch(config-mst)# revision 1
```

Homing the bridge within a region on SMB Switch - GUI

The MSTP Properties page is used to define what region the switch is in. For devices to be in the same region, they must have the same region name and revision value.

Step 1. Choose **Spanning Tree > MSTP Properties** from the menu.

Step 2. Enter a **name** for the MSTP region in the *Region Name* field. The region name defines the logical boundary of the network. All switches in an MSTP region must have the same configured region name.

Step 3. Enter a **revision number** in the *Revision field*. This is a logical number that signifies a revision for the MSTP Configuration. All switches in an MSTP region must have the same revision number.

Step 4. Enter the maximum number of **hops** in the *Max Hops* field. Max Hops specifies the lifetime of BPDUs in hop counts. When a bridge receives a BPDU, it decrements the hop count by one and resends the BPDU with the new hop count. Once a bridge receives a BPDU with a hop count of zero, the BPDU is discarded.

Note: The *IST Active* field displays the bridge priority and MAC address of the active switch of the region. [See glossary for additional information.](#)

Step 5. Click **Apply**.

Best Practice #7 - Nest the root bridge of the CIST within the primary MST region

This best practice is part of the lynchpin to hold the whole migration together. The idea is to place the root bridge for the MSTP topology - within the primary MSTP region. Given the previous best practice placing all VLANs within the same region, root selection is valid for all VLANs. This is achieved via the feature known as Root Guard, which enforces root placement created by you. When a bridge receives a superior BPDU on a root guard activated port, it will immediately place the port in listening mode, via root-inconsistent STP state. This prevents forwarding of their inferior BPDUs, thereby preserving the designated ports on the root bridge of your region. Thereby preserving the designated ports on the root bridge of your region.

Note: Carefully select the root and a back-up root for each instance.

Placing the root bridge onto the CIST on Catalyst Switch - CLI

```
Catalyst(config)# spanning-tree mst [instance-id] root {primary | secondary} [diameter dia  
[hello-time hello-time]]
```

```
Catalyst(config)# spanning-tree mst 1 root primary 7
```

Troubleshoot - Catalyst

The following command will return any ports that have been marked inconsistent. But also note the command is not available on SMB switches.

```
Catalyst# show spanning-tree inconsistentports
```

Placing the root bridge onto the CIST on SMB Switch - CLI

```
SMBswitch(config)# interface [interface-id]  
SMBswitch(config-if)# spanning-tree guard root
```

```
SMBswitch(config)# interface gi1/1/1  
SMBswitch(config-if)# spanning-tree guard root
```

Placing the root bridge onto the CIST on SMB Switch - GUI

Step 1. Log in to the web configuration utility and choose **Spanning Tree > STP Interface Settings**.

Step 2. Choose an **interface** from the *Interface Type* drop-down list.

Step 3. Click **Go** to display a list of ports or LAGs on the interface.

Step 4. Click the **radio button** of the port *or* **LAG** you want to modify and click **Edit**. The Edit STP Interface Setting window appears.

Step 5. Click the **radio button** that corresponds to the desired interface in the Interface field.

- Port — From the Port drop-down list, choose the port to configure. This will only affect the single port chosen.
- LAG — From the LAG drop down list, choose the LAG to configure. This will affect the group of ports defined in the LAG configuration.

Step 6. Ensure STP is checked **Enable** in the *STP* field to enable STP on the interface.

Step 7. Check **Enable** in the *Root Guard* field to enable Root Guard on the interface. This option provides a way to enforce the root bridge placement in the network. Root Guard is used to prevent a new connected device to take over as root bridge.


Migration Verification - Is this thing on?

At this point, your MSTP implementation and network should be purring along. For the trust-but-verify crowd, you're able to verify MSTP status by performing a frame capture. Then compare the results to your expected documentation.

After performing a packet capturing via Wireshark, you will see *Mrecords* that contain the instance id. Below is a screenshot of the *Mrecord*, prior to expansion for additional detail.

▼ Spanning Tree Protocol

```
Protocol Identifier: Spanning Tree Protocol (0x0000)
Protocol Version Identifier: Multiple Spanning Tree (3)
BPDU Type: Rapid/Multiple Spanning Tree (0x02)
▶ BPDU flags: 0x7c, Agreement, Forwarding, Learning, Port Role: Designated
▶ Root Identifier: 24576 / 0 / 24:e9:b3:78:fe:80
Root Path Cost: 0
▶ Bridge Identifier: 24576 / 0 / 24:e9:b3:78:fe:80
Port identifier: 0x8018
Message Age: 0
Max Age: 20
Hello Time: 2
Forward Delay: 15
Version 1 Length: 0
Version 3 Length: 96
▼ MST Extension
MST Config ID format selector: 0
MST Config name: Cisco
MST Config revision: 1
MST Config digest: 2a5477095c475f337a69c797b32cd60a
CISt Internal Root Path Cost: 0
▶ CISt Bridge Identifier: 24576 / 0 / 24:e9:b3:78:fe:80
CISt Remaining hops: 20
▶ MSTID 1, Regional Root Identifier 24576 / 24:e9:b3:78:fe:80
▶ MSTID 2, Regional Root Identifier 24576 / 24:e9:b3:79:06:00
```



Expanding the *Mrecord* Allows you to view more granular data about MSTP. Including:

- Port Role
- MST ID
- Regional Root
- Internal Path Cost
- Bridge Identifier Priority
- Port Identifier Priority
- Remaining Hops

```
▼ MSTID 1, Regional Root Identifier 24576 / 24:e9:b3:78:fe:80
▶ MSTI flags: 0x7c, Agreement, Forwarding, Learning, Port Role: Designated
0110 .... = Priority: 0x6
.... 0000 0000 0001 = MSTID: 1
Regional Root: Cisco_78:fe:80 (24:e9:b3:78:fe:80)
Internal root path cost: 0
Bridge Identifier Priority: 6
Port identifier priority: 8
Remaining hops: 20
▼ MSTID 2, Regional Root Identifier 24576 / 24:e9:b3:79:06:00
▶ MSTI flags: 0x78, Agreement, Forwarding, Learning, Port Role: Root
0110 .... = Priority: 0x6
.... 0000 0000 0010 = MSTID: 2
Regional Root: Cisco_79:06:00 (24:e9:b3:79:06:00)
Internal root path cost: 20000
Bridge Identifier Priority: 8
Port identifier priority: 8
Remaining hops: 20
```

Quick verification commands - SMB CLI

If you'd like to verify from the command line, try these commands:

```
SMBswitch# show spanning-tree mst-configuration

SMBswitch(config)# spanning-tree mst-configuration
SMBswitch(config-mst)# show pending
```

```
Pending MST configuration
Name [region1]
Revision 1
Instances configured 2
Instance Vlans Mapped
```

```
-----
0 1-9,21-4094
1 10-20
-----
```

```
SMBswitch# show spanning-tree mst-configuration
Name []
Revision 0
Instance Vlans mapped
```

```
-----
0 1-4094
-----
```

Note:The Catalyst version of the show command excludes the - between mst and configuration.
EX:"show spanning-tree mst configuration"

What to know about PVST+ and MSTP living on the same network

If you need to continue support for legacy switches running PVST+, handle this on a port by port basis. If one of these switches runs as a VLAN trunk, ensure the MSTP switch is the root for all VLANS assigned to the trunk. Further, MSTP attempts to decode PVST+ BPDUs but this simulation is imperfect. Which requires us to dive into the idea of Boundary Ports.

The role and state of a MSTP boundary port is determined by the *Internal Spanning Tree* interacting with exterior topology. This means that if a port is in blocking mode on the *IST*, then it is blocking in all instances of MSTP. This effect cascades into the PVST+ implementation, affecting the VLANs function. The same holds true if the port is forwarding, learning, etc. As you might imagine, this can become an issue. This can result in an intractable issue whereas a port that should be forwarding for one VLAN, is instead blocking, due to the needs of another VLAN. The PVST+ simulation leverages information from the *IST* to create per VLAN BPDUs. This results in a network wide "illusion" that the MSTP region appears as a single switch to all the VLANs. Similar to the way switches are able to *stack*, which isn't half bad. What is bad, from the boundary port's position, is that it creates the need to send individual BPDUs for each simulated VLAN. Any inconsistency between BPDUs can tear down the whole simulation in errors. Only receiving consistent BPDUs will allow the simulation to stand itself back up.

To conclude, this whole situation is the reason why the BPDUs received on the boundary port must be identical. [For additional reading on this topic, reference this community thread.](#)

Is there anything to know, if my network hardware...isn't entirely Cisco?

MSTP is backwards compatible. As long as your non-Cisco hardware supports Rapid Spanning Tree, you should be alright. If you run into issues, [check with our switching community.](#)

Conclusion

Thanks for reading through this guide, with these best practices you should be set to improve performance of your layer-2 network.

Worth noting, spanning tree may not sound thrilling to you, but the benefits of load sharing make it worth the effort for keeping your network efficient. The creator of spanning tree, Radia Perlman, loves it as much as a mother ever could. She even wrote a [book](#) about it.