Cisco AVVID Network Infrastructure: Implementing 802.1w and 802.1s in Campus Networks

Implementation Guide
April, 2003

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: 956652

CONTENTS

About this Guide

This document presents an overview of Rapid Spanning-Tree Protocol (RSTP) and Multiple Spanning-Tree Protocol (MSTP) and how to implement each.

Intended Audience

This document is an implementation guide for deploying the recently ratified 802.1w (RSTP) and 802.1s (MSTP) in enterprises where Layer 2 redundancy is required and spanning tree is used to prevent Layer 2 loops.

This document includes an over view of RSTP and MSTP as well as configuration examples, implementation details, and a discussion of interoperability issues with legacy spanning tree.

Document Organization

This document contains the following chapters:

Chapter or Appendix	Description
Chapter 1, “Introduction”	Provides an introduction for this implementation guide.
Chapter 2, “Understanding Rapid Spanning-Tree Protocol (802.1w)”	Provides an overview of the RSTP (802.1w).
Chapter 3, “Understanding Multiple Spanning-Tree Protocol (802.1s)”	Provides an overview of the MSTP (802.1s).
Chapter 4, “Deploying RSTP and MSTP”	Provides guidelines and examples for implementing RSTP and MSTP.

Document Conventions

This guide uses the following conventions to convey instructions and information:

Table 1 Document Conventions

Convention	Description
boldface font	Commands and keywords.
italic font	Variables for which you supply values.
[ ]	Keywords or arguments that appear within square brackets are optional.
{x | y | z}	A choice of required keywords appears in braces separated by vertical bars. You must select one.
`screen font`	Examples of information displayed on the screen.
`boldface screen font`	Examples of information you must enter.
< >	Nonprinting characters, for example passwords, appear in angle brackets.
[ ]	Default responses to system prompts appear in square brackets.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Tips Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Obtaining Documentation

These sections explain how to obtain documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com

Translated documentation is available at this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which is shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Ordering Documentation

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. In the Cisco Documentation home page, click the **Fax** or **Email** option in the "Leave Feedback" section at the bottom of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit your comments by mail by using the response card behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

If you want to obtain customized information and service, you can self-register on Cisco.com. To access Cisco.com, go to this URL:

http://www.cisco.com

Technical Assistance Center

The Cisco Technical Assistance Center (TAC) is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Cisco TAC inquiries are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

The Cisco TAC resource that you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

You can use the Cisco TAC Web Site to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://www.cisco.com/register/

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC Web Site, you can open a case online by using the TAC Case Open tool at this URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

Introduction

This chapter provides an overview of the situations where Layer 2 loops may exist in campus networks and the tools that are available to address these loops.

Hierarchical Campus Networks

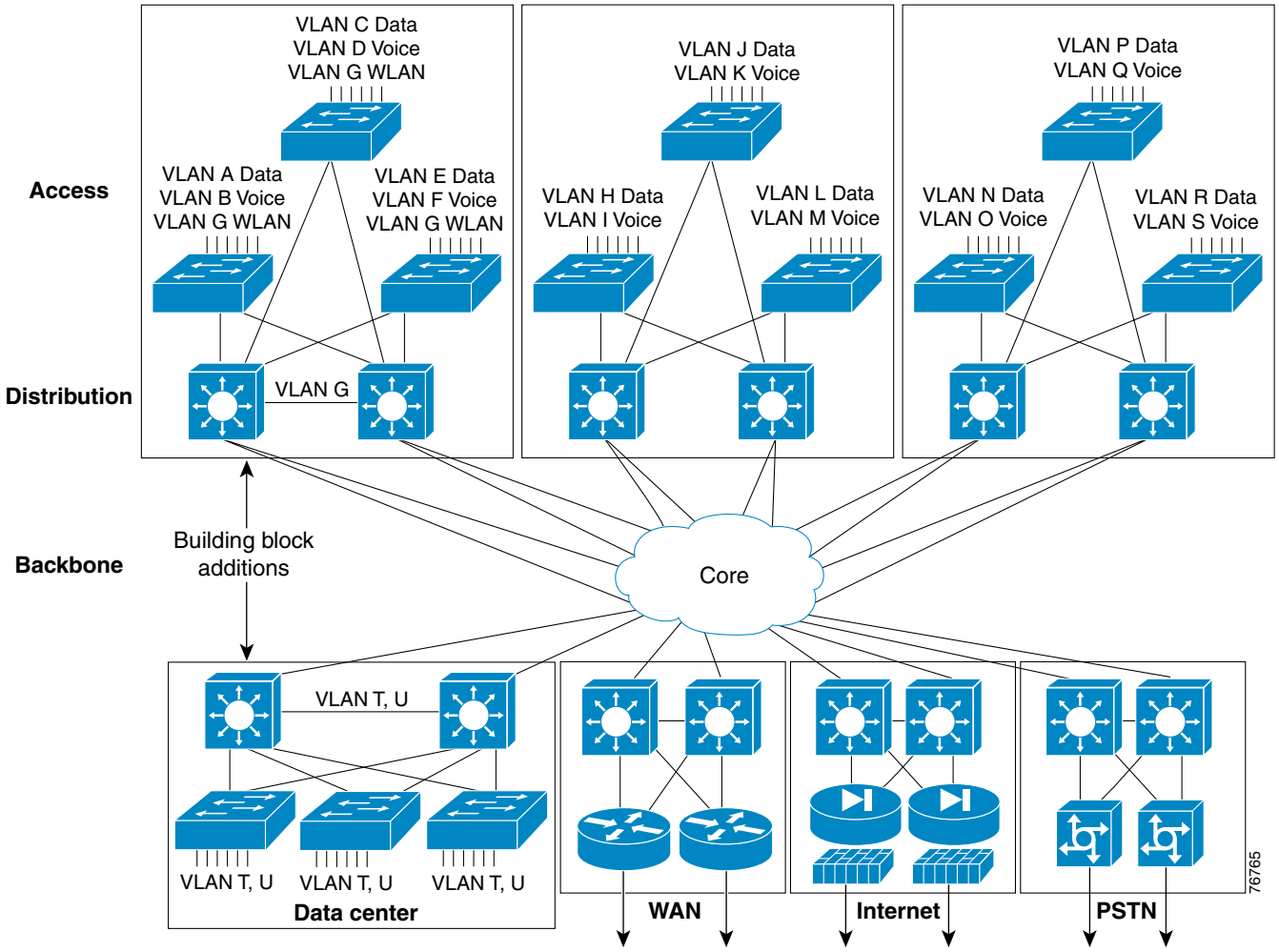
For campus networks, Cisco recommends a hierarchical network design that distributes networking function at each layer through a layered organization. The hierarchical model enables the design of a modular topology using "building blocks" that are scalable and allow the network to meet evolving business needs.

The hierarchical model is based on a modular design, which is easy to scale, understand, and troubleshoot because it follows a deterministic traffic pattern. The principle advantages of the hierarchical model are:

- Hierarchy—With a hierarchical design, flows get larger as they traverse points of aggregation and move up the hierarchy. Functions are distributed at each layer in an optimal way through a layered organization. And a hierarchical design avoids the need for a fully meshed network, in which all devices are connected to each other. This promotes scalability.
- Modularity—Modular networks are made from building blocks, which are easy to replicate, redesign, and grow. Each time a module is added or taken out, there should be no need to redesign the whole network. Distinct blocks can be put in-service and taken out-of-service without impacting other blocks or the core of the network. This greatly enhances the ease of troubleshooting, problem isolation, and network management.

Cisco introduced the hierarchical design model in 1999. This model, shown in Figure 1-1, uses a layered approach with the primary components being the access layer, the distribution layer, and the core (backbone) layer. Server farms (data centers), WANs, Internet connections, and PSTNs can be plugged in as building blocks in this model.

Figure 1-1 Hierarchical Campus Network Design



Cisco recommends that campus designs avoid the use of Layer 2 loops whenever possible. With the advent of hardware-accelerated, Layer 3 switches, which offer intelligent network services (INS) and routing at Layer 2 switching rates, there are few reasons to extend a Layer 2 domain across campus. However, there are two situations in which Layer 2 loops might be unavoidable and a Spanning Tree Protocol (STP) must be used. These situations include the use of:

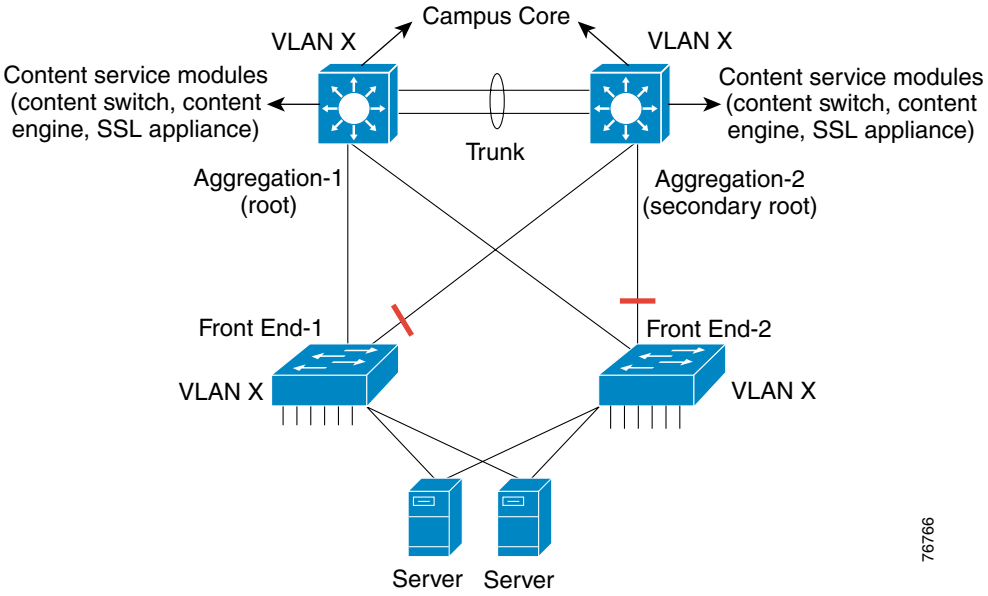
- Data Centers
- Wireless LANs

Data Centers

A data center houses server farms, which consist of a logical group of networked servers. These servers are tasked with handling various processes, like those of web, application, and database services. Server farms often take advantage of other infrastructure devices, such as Content Switches, Content Engines, and Secure Sockets Layer (SSL) appliances, to assist in offloading the processing requirements of individual servers and the server farm as a whole.

Designing a data center is different from designing a standard Distribution-Access layer block for end users. When using dual-homed servers, it is necessary to have the same Layer 2 VLAN appear in multiple devices (as shown in Figure 1-2). This means that a Spanning Tree, such as 802.1D or 802.1w, is required to create a loop-free Layer 2 topology.

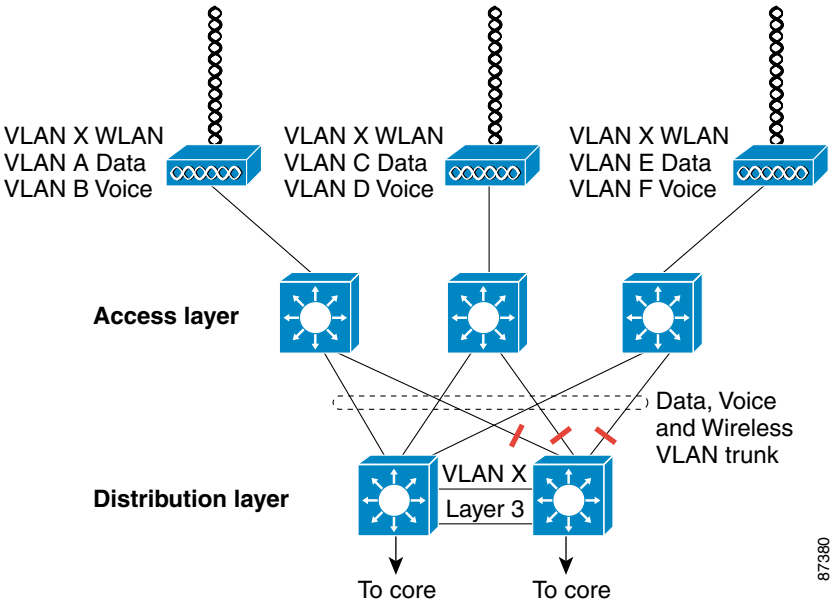
Figure 1-2 Data Centers and Layer 2 Loops



Wireless LANs

Wireless LANs enable users to connect to a network from any location within an enterprise. However, moving (or roaming) from one wireless access point (AP) to another without being dropped is not possible if the APs are in different IP subnets—unless Mobile IP is used. To avoid the complexities of Mobile IP, APs can be located in within the same Layer 2 domain, or VLAN, to provide the fastest roaming time for mobile end stations. This means that the wireless VLAN must exist in the entire building or even an entire campus. Because the devices in the wiring closet should be redundantly connected for availability, this introduces the likelihood of Layer 2 loops (as shown in Figure 1-3). Therefore, a Spanning Tree, such as 802.1D or 802.1w, is required to create a loop-free Layer 2 topology.

Figure 1-3 WLAN s and Layer 2 Loops



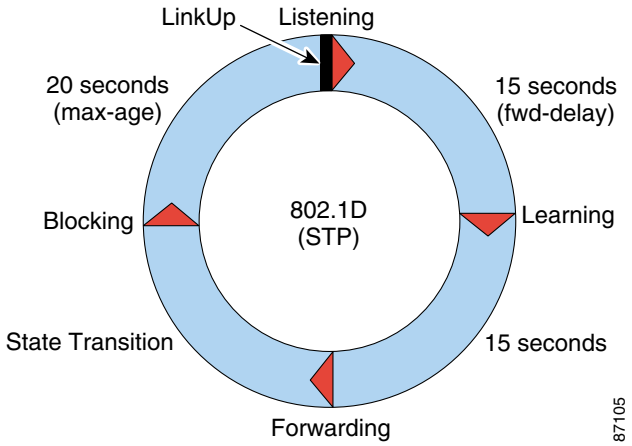
Spanning Tree Evolution

For many years, STP existed as an unenhanced standard. In recent years, however, the protocol has seen many enhancements and changes.

802.1D

Initially, redundant switched networks had to rely on the relatively sluggish 802.1D STP to address the problems of Layer 2 loops. The 802.1D standard was designed by the Institute of Electrical and Electronics Engineers (IEEE) at a time where recovering connectivity (and cycling through the STP states, shown in Figure 1-4) after an outage within a minute or so was considered adequate performance. As the tolerance level for outages reduced, maintaining 802.1D often turned out to be the network administrator's most challenging task, as tuning the protocol timers was the only way to get a few seconds of faster convergence, but often to the detriment of the network's stability.

Figure 1-4 802.1D Cycle



Cisco 802.1D Enhancements

In the late 1990s, Cisco enhanced the original 802.1D specification with features such as UplinkFast, BackboneFast, and PortFast to speed up the convergence time of a bridged network.

- UplinkFast is an access-layer STP solution that provides fast failover when the root port or root switch fails.
- BackboneFast is a distribution and access-layer STP solution that provides fast convergence in the network for indirect link failures.
- PortFast is an access-layer STP solution that causes a port to enter the spanning tree forwarding state immediately, bypassing the listening and learning states.

The drawback of these mechanisms is that they are proprietary and require additional configuration. Cisco also answered the scalability issues of Layer 2 based networks by developing the Multiple Instance Spanning Tree Protocol (MISTP).

Rapid and Multiple Spanning Tree

In 1999, the IEEE decided to incorporate most of these concepts into two standards, which were ratified in 2002: Rapid Spanning-Tree Protocol (RSTP; 802.1w) and Multiple Spanning-Tree Protocol (MSTP; 802.1s). Using these new protocols, convergence times in the hundreds of milliseconds can be expected while scaling to thousands of VLANs.

Cisco remains the leader in the industry by offering these two protocols in addition to the proprietary STP enhancements (discussed in the previous section) to facilitate the migration and interoperability with legacy bridges.

The remainder of this document provides an overview of the new spanning-tree protocols and how they should be implemented in enterprises that use the multilayer design model.

CHAPTER 2

Understanding Rapid Spanning-Tree Protocol (802.1w)

Rapid Spanning-Tree Protocol (RSTP; IEEE 802.1w) is an evolution of the IEEE 802.1D standard. RSTP is a Layer 2 loop prevention algorithm like 802.1D. However, RSTP achieves rapid failover and convergence times in many situations, such as switch failure, cable failure, and topology change for Layer 2 networks.

RSTP is not a timer-based spanning tree algorithm (STA) like 802.1D. Therefore, RSTP offers an improvement over the 30 seconds or more that 802.1D takes to move a link to forwarding. The heart of the protocol is a new bridge-bridge handshake mechanism, which allows ports to move directly to forwarding.

RSTP is now the recommended STA for resilient networks relying on Layer 2 cable paths for redundancy. It is backwardly compatible with 802.1D, transparent to end users, and more importantly standards based. Some of the enhancements in RSTP are achieved through the introduction of:

- New port role assignments and port states
- New BPDU format and BPDU processing
- A bridge-bridge handshake mechanism, which rapidly determines protocol state for the link
- A different Topology Change Notification and processing procedure

The 802.1D terminology remains primarily the same and most parameters have been left unchanged. Therefore, users familiar with 802.1D can quickly and easily configure the new protocol. 802.1w is also capable of reverting back to 802.1D in order to interoperate with legacy bridges (thus dropping the benefits it introduces) on a per-port basis.

This chapter provides an overview of the enhancements added by RSTP to the previous 802.1D standard.

Note RSTP was first implemented as part of Multiple Spanning-Tree Protocol (MSTP) in Catalyst OS 7.1 and IOS software release 12.1(11)EX and later. It is currently available as a standalone protocol with the Rapid Per-VLAN-Spanning-Tree Plus (Rapid-PVST+) mode on the Catalyst 6000 in IOS software release 12.1(13)E and Catalyst OS 7.4, on the Catalyst 3550 switch in IOS software release 12.1(13)EA1, and on the Catalyst 4000 in Catalyst OS 7.4. In this mode, the switch runs an RSTP instance on each VLAN.

New Port States and Port Roles

802.1D defined four different port states: listening, learning, blocking, and forwarding. This was a bit confusing because it mixed the state of a port (whether it blocks or forwards traffic) and the role it plays in the active topology (root port, designated port, and so on). For example, from an operational point of view, there is no difference between a port in blocking state and a port in listening state; they both discard frames and do not learn MAC addresses. The real difference lies in the role that the spanning tree assigns to the port. It can safely be assumed that a listening port is either designated or root and is on its way to the forwarding state. Unfortunately, once in forwarding state, there is no way to infer from the port state whether the port is root or designated. RSTP addresses this confusion by decoupling the role and the state of a port.

Note RSTP calculates the final topology for the spanning tree using the same criteria as 802.1D. There is no change in the way the different bridge and port priorities are used.

Port States

There are only three port states in RSTP, which correspond to the three possible operational states.

- Learning
- Forwarding
- Discarding

The 802.1D states disabled, blocking, and listening have been merged into a unique 802.1w discarding state.

Note In the Cisco implementation, the name *blocking* is used for the discarding state. Catalyst OS release 7.1 and later still display the listening and learning states, giving even more information about a port than the IEEE standard requires. However, there now is a difference between the role the protocol has determined for a port and its current state. For example, it is now perfectly valid for a port to be designated and blocking at the same time. While this will typically happen for very short periods of time, it simply means that this port is in a transitory state towards designated forwarding.

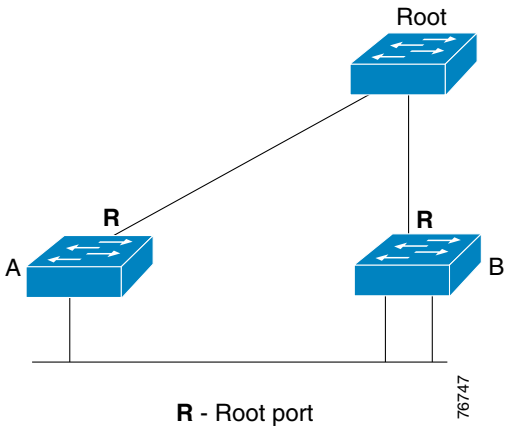
Port Roles

The role is now a variable assigned to a given port. The *root port* and *designated port* roles remain. The blocking port role is now split into the *backup* and *alternate port roles*. The STA determines the role of a port based on an examination of the Bridge Protocol Data Units (BPDUs) to decide whether one is more useful than the other. This decision is based on the value stored in the BPDU (and occasionally on the port on which they are received). The value of the BPDU then determines the role of the port, as explained in the following sections.

Root Port Roles

With STP, the STA elects a single root bridge for the whole bridged network (per-VLAN). The root bridge sends BPDUs that are more useful than the ones that any other bridge can send. The port receiving the best BPDU on a bridge is the *root port*. This is the port that is the closest to the root bridge in terms of path cost.

Figure 2-1 Root Port

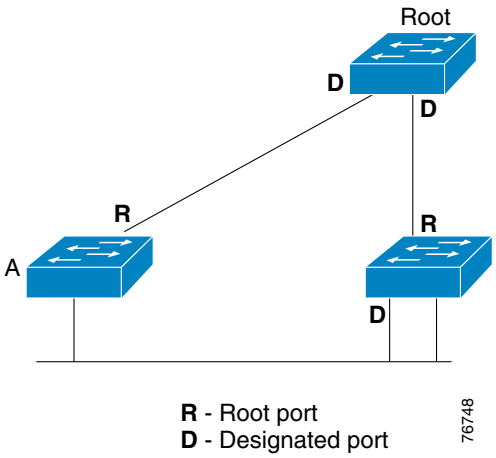


Note The root bridge is the only bridge in the network that does not have a root port. All other bridges receive BPDUs on at least one port.

Designated Port Role

802.1D bridges create a bridged domain by linking together different segments (Ethernet segments, for example). On any given segment, there can be only one path toward the root bridge. If there were two, there would be a bridging loop in the network. All bridges connected to a given segment listen to each other's BPDUs and agree on the bridge sending the best BPDU as the designated bridge for the segment. The corresponding port on that bridge is the *designated port*.

Figure 2-2 Designated Port



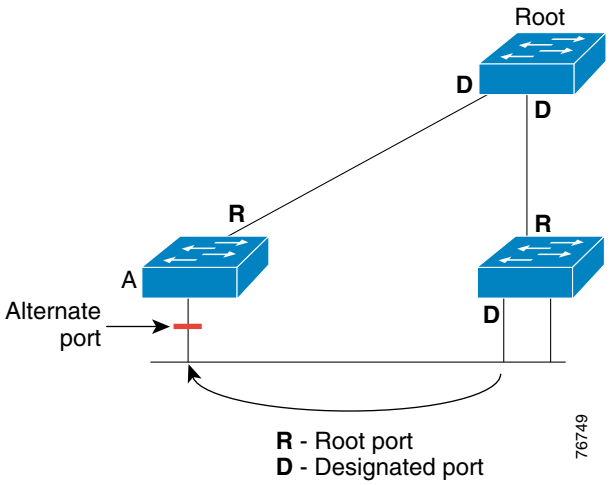
Alternate and Backup Port Roles

These two port roles correspond to the blocking state of 802.1D. A blocked port is defined as any port that is not the designated or root port. A port remains blocked as long as it receives more useful BPDUs than the one it would send out on its segment. Therefore, a port must receive BPDUs in order to stay blocked.

With RSTP, there are two types of blocked ports.

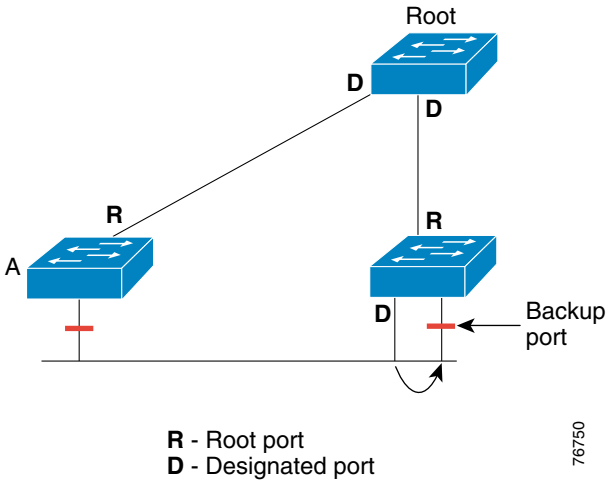
- An *alternate port* is a port that is blocked because it is receiving more useful BPDUs from another bridge, as shown in Figure 2-3.

Figure 2-3 Alternate Port



- A *backup port* is a port that is blocked because it is receiving more useful BPDUs from the same bridge it is on, as shown in Figure 2-4.

Figure 2-4 Backup Port



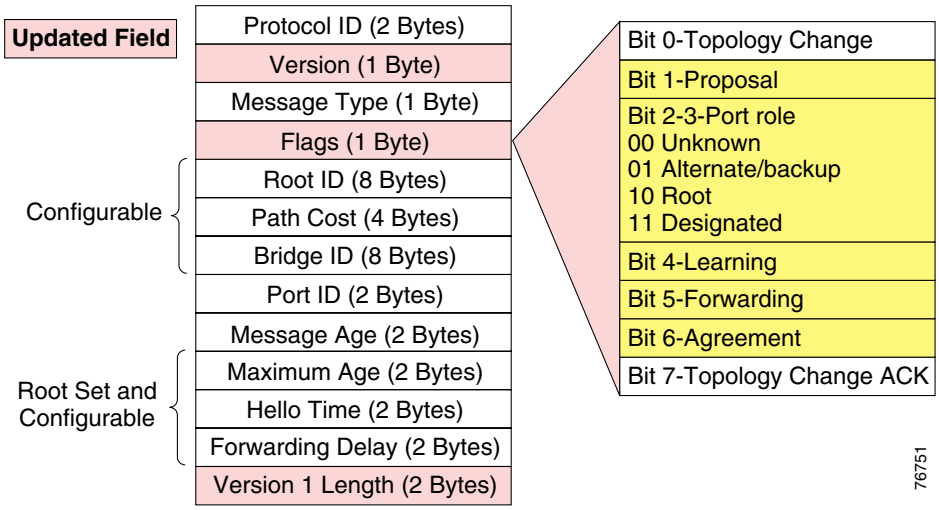
This distinction was already made internally within 802.1D and this is essentially how Cisco's UplinkFast feature functions. The rationale behind this is that an alternate port provides an alternate path to the root bridge and could, therefore, replace the root port should it fail. A backup port provides redundant connectivity to the same segment and cannot guarantee an alternate connectivity to the root bridge.

New BPDU Format

A few changes to the BPDU format have been introduced by RSTP (as shown in Figure 2-5). Only two flags, Topology Change (TC) and TC Acknowledgment (TCA), were defined in 802.1D. However, RSTP now uses the six remaining bits of the flag byte to do the following:

- Encode the role and state of the port from which the BPDU originated
- Handle the proposal/agreement mechanism

Figure 2-5 RSTP BPDU Format



Another important change is that the RSTP BPDU is now of Type 2, Version 2. The implication of this is that legacy bridges must drop this new BPDU. This makes it easy for a 802.1w bridge to detect the legacy bridges connected to it. RSTP BPDUs are of Type 2, Version 2 and MST BPDUs are Type 2, Version 3 format.

Note BPDUs are sent to the same IEEE MAC address.

New BPDU Handling

With 802.1D, a non-root bridge only generates BPDUs when it receives one on its root port. In fact, with 802.1D, a bridge relays BPDUs instead of generating them.

With 802.1w, a bridge sends a BPDU with its current information at the hello time interval (2 seconds by default), even if it does not receive any BPDUs from the root bridge.

Faster Aging of Information

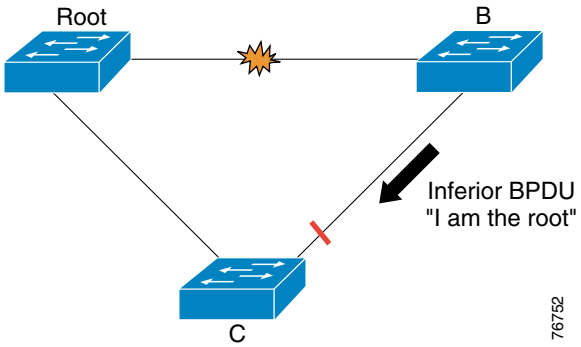
In 802.1D on any given port, if BPDUs are not received before the max_age timer (20 seconds, by default) expires, the protocol information is aged out. With 802.1w, BPDUs are now used as a keep-alive mechanism between bridges. If a bridge misses three BPDUs in a row, it considers the connection to its direct neighboring root or designated bridge to be lost and immediately ages out the protocol information. This is in contrast to 802.1D where the problem could have been anywhere on the path to the root. This fast aging of the information allows for quick failure detection.

Note Failures are detected even faster in the case of a physical link failure.

Accepting Inferior BPDUs

Accepting inferior BPDUs is what makes up the core of the BackboneFast feature. The IEEE 802.1w committee decided to incorporate a similar mechanism into RSTP. When a bridge receives inferior information from its designated or root bridge, it immediately accepts it and replaces the one previously stored, as shown in Figure 2-6.

Figure 2-6 Inferior BPDUs



Because Bridge C still knows the root is alive and well, it immediately sends a BPDU to Bridge B containing information about the root bridge. As a result, Bridge B stops sending its own BPDUs and accepts the port leading to Bridge C as its new root port.

Rapid Transition to Forwarding State

Rapid transition is the most important feature introduced by 802.1w. The legacy STA passively waited for the network to converge before moving a port into the forwarding state. Achieving faster convergence was a matter of tuning the conservative default parameters (forward delay and max_age timers), often sacrificing the stability of the network.

RSTP is able to actively confirm that a port can safely transition to forwarding without relying on any timer configuration. There is a feedback mechanism that operates between RSTP-compliant bridges. To achieve fast convergence on a port, the RSTP relies on two new variables: edge ports and link type.

Edge Ports

The *edge port* concept is already well known to Cisco's spanning tree users as it basically corresponds to the PortFast feature. The idea is that ports that are directly connected to end stations cannot create bridging loops in the network and can thus directly transition to forwarding, skipping the listening and learning stages. An edge port does not generate topology changes when its link toggles. Unlike PortFast though, an edge port that receives a BPDU immediately loses its edge port status and becomes a normal spanning-tree port. At this point, there is a user-configured value and an operational value for the edge port state.

In Cisco's implementation, the **portfast** command is used for edge port configuration, thus making the transition to RSTP simpler.

For more information about the portfast command, see the “PortFast” section on page 4-23.

Link Type

RSTP can only achieve rapid transition to forwarding on edge ports and on point-to-point links. The link type is automatically derived from the duplex mode of a port. A port operating in full-duplex will be assumed to be point-to-point, while a half-duplex port will be considered as a shared port by default. This automatic link type setting can be overridden by explicit configuration.

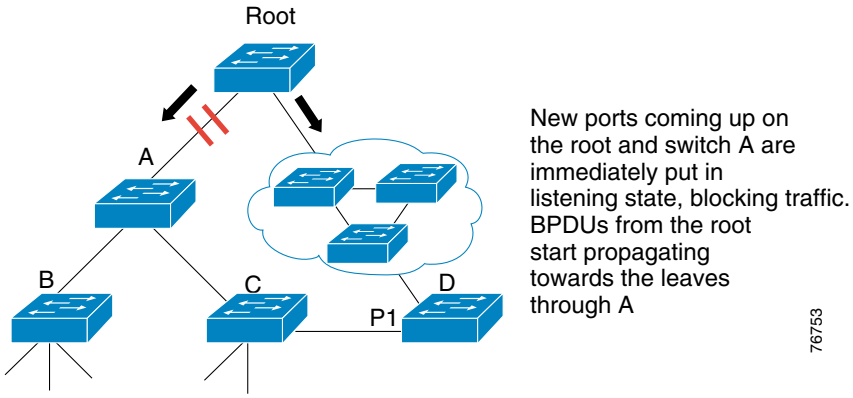
In today's switched networks, most links are operating in full-duplex mode and are therefore treated as point-to-point links by RSTP. This makes them candidates for rapid transition to forwarding.

Convergence in 802.1D

In this scenario, a link between the root bridge and Bridge A has just been added. Let’s assume there was already an indirect connection between Bridge A and the root bridge (via Bridge C to Bridge D in the diagram). The STA disables the bridging loop by blocking a port.

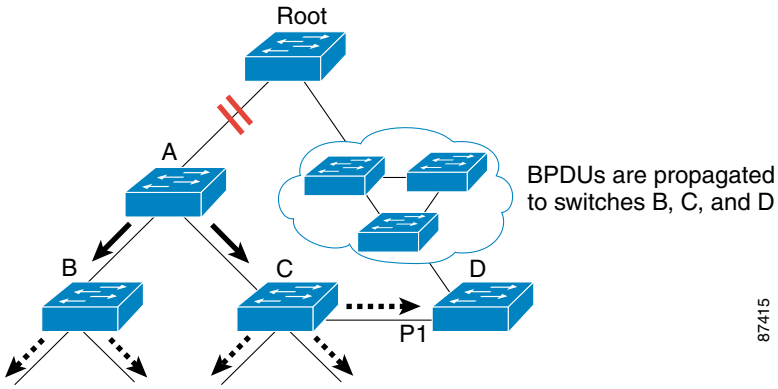
1. As they are just coming up, both ports on the link between the root and A are put in listening state. Bridge A is now able to hear the root directly and it immediately propagates its BPDUs on its designated port (as shown in Figure 2-7).

Figure 2-7 Convergence with 802.1D—Initial Step



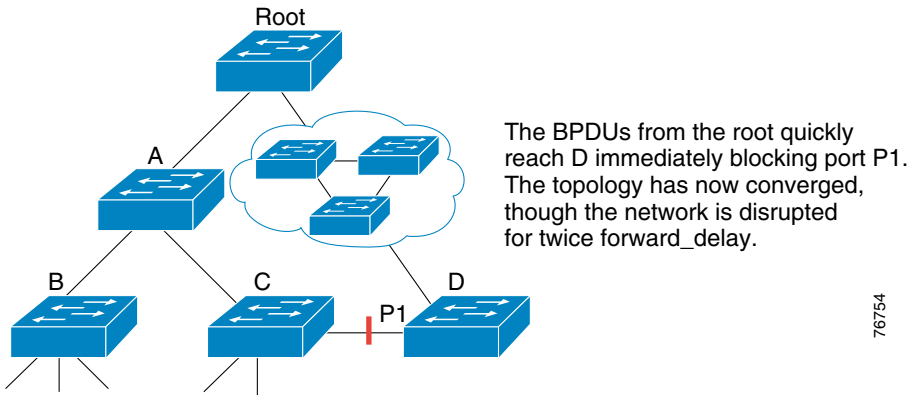
2. As soon as Bridge B and Bridge C receive this new BPDU with indicating a superior path cost from Bridge A, they immediately relay it toward the leaves (as shown in Figure 2-8).

Figure 2-8 Convergence with 802.1D—Interim Step



3. In a few seconds, Bridge D has received a BPDU from the root and instantly blocks its port P1 (as shown in Figure 2-9).

Figure 2-9 Convergence in 802.1D—Final Topology



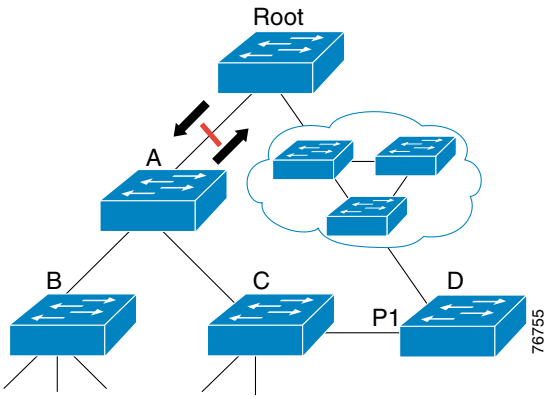
Spanning tree has been very efficient in calculating the new topology of the network. The only problem now is that twice the forward delay has to elapse before the link between the root and Bridge A eventually ends up in the forwarding state. This means 30 seconds of disruption of traffic (the entire Bridge A, Bridge B, and Bridge C part of the network is isolated) because the 8021.d algorithm lacks a feedback mechanism to clearly advertise that the network has converged in a matter of seconds.

Convergence in RSTP

Using the same example, let's examine how RSTP (802.1w) deals with a new link. Remember that the final topology is exactly the same as the one calculated by 802.1D (that is, one blocked port at the same place as before), only the steps used to reach this topology have changed.

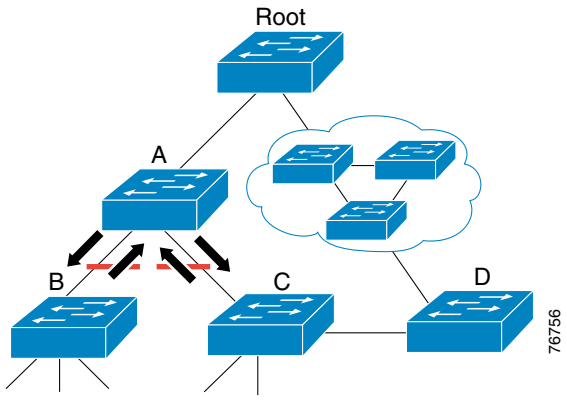
1. Both ports on the link between Bridge A and the root bridge are put in designated blocking as soon as they come up (as shown in Figure 2-10).

Figure 2-10 Convergence in 802.1w—Root Bridge Blocked



2. At this point, a negotiation takes place between Bridge A and the root. As soon as A receives the root's BPDU, it blocks its non-edge designated ports. This operation is called *sync*. Once Bridge A has blocked its non-edge designated ports, the link between Bridge A and the root bridge is put in forwarding state (as shown in Figure 2-11).

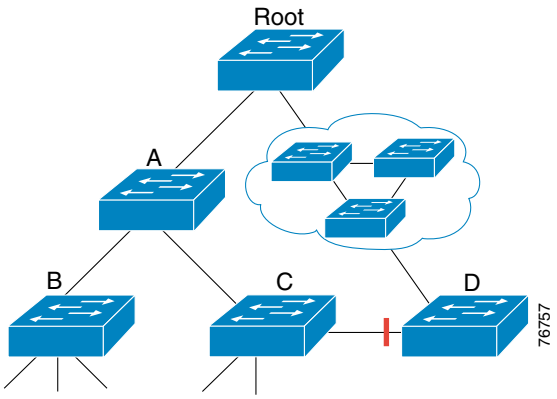
Figure 2-11 Convergence in 802.1w—Root Bridge Forwarding



There is still no loop because instead of blocking above Bridge A, the network is now blocking below Bridge A. The potential bridging loop is cut, however at a different location. This cut is traveling down the tree along with the new BPDUs originated by the root traveling through Bridge A. At this point, the newly blocked ports on Bridge A will also negotiate a quick transition to forwarding with their neighboring ports on Bridge B and Bridge C. Bridge B has only edge designated ports (other than its root port towards A), so it has no port to block in order to authorize Bridge A to go to forwarding.

3. Bridge C had to only block (sync) its designated port to Bridge D.

Figure 2-12 Convergence in 802.1w—Final Topology



The final topology is exactly the same as for the 802.1D example, which means that port P1 on Bridge D is blocking. The final network topology has been reached in just the time necessary for the new BPDUs to travel down the tree. No timer is been involved in this quick convergence. The only new mechanism introduced by RSTP is the acknowledgment that a switch can send on its new root port in order to authorize immediate transition to forwarding, bypassing the twice-the-forward-delay long listening and learning stages.

To benefit from fast convergence, the administrator needs only to remember the following:

- This negotiation between bridges is only possible when bridges are connected by point-to-point links (that is, full-duplex links unless explicit port configuration).
- Edge ports play an even more important role now that portfast is enabled on ports in 802.1D. If the network administrator failed to properly configure the edge ports on Bridge B, for example, their connectivity would have been impacted by the link between Bridge A and Bridge B coming up because the ports would have to go through the standard sync process.

Proposal/Agreement Handshake Sequence

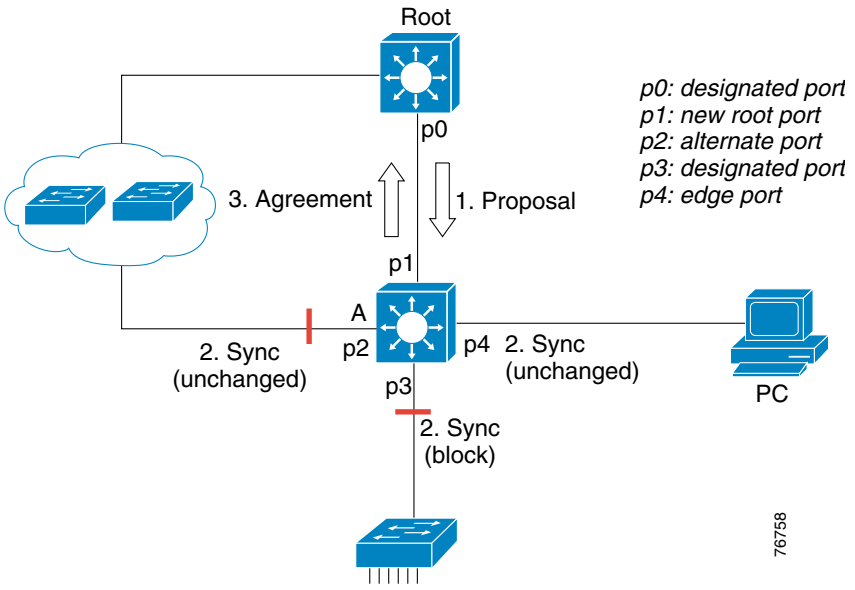
There is no feedback or explicit bridge to bridge communication in 802.1D. Therefore, when a port has been selected by the STA to become a designated port, 802.1D still waits twice the designated forward delay (30 seconds by default) before transitioning it to the forwarding state. In RSTP, this condition corresponds to a port with a designated role but a blocking state.

RSTP, however, uses a proposal/agreement handshake sequence to quickly transition the ports to achieve fast convergence. The following example illustrates how the quick transition is achieved.

In this example, a new link has been created between the root bridge and Bridge A. Both ports on this link are put in a designated blocking state until they receive a BPDU from their counterpart.

Step 1 After the link comes up, port p0 and p1 are put into designated blocking and both send an RSTP BPDU with the proposal bit set. Because Bridge A receives a BPDU indicating a superior path cost, it immediately knows that p1 is going to be its new root port.

Figure 2-13 Proposal/Agreement Sequence



Step 2 Bridge A then starts a sync to ensure that all of its ports are in sync with this new information. A port is in sync if it meets either of the following criteria:

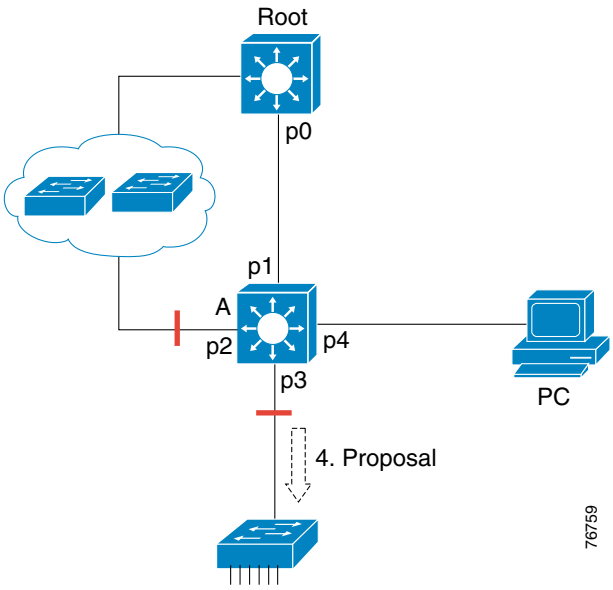
- It is in a blocking state (which means discarding, in a stable topology).
- It is an edge port.

In this example, there is an alternate port p2, a designated forwarding port p3, and an edge port p4 on Bridge A. Notice that p2 and p4 already meet one of the criteria listed above. To be in sync, Bridge A must block port p3, assigning it to the discarding state.

Step 3 Now that all of its ports are in sync, Bridge A can now unblock its newly selected root port p1 and reply to the root by sending an agreement message. This is a copy of the proposal BPDU, with the agreement bit set instead of the proposal bit. This ensures that port p0 knows which proposal corresponds to the agreement it receives.

Step 4 Once p0 receives that agreement, it can immediately transition to forwarding (as shown in Figure 2-14).

Figure 2-14 Proposal BPDU



Notice that port p3 was left in a designated discarding state during the sync. In Step 4, that port is in the exact same situation as port p0 was during Step 1. Port p3 then starts proposing to its neighbor, attempting to quickly transition to forwarding.

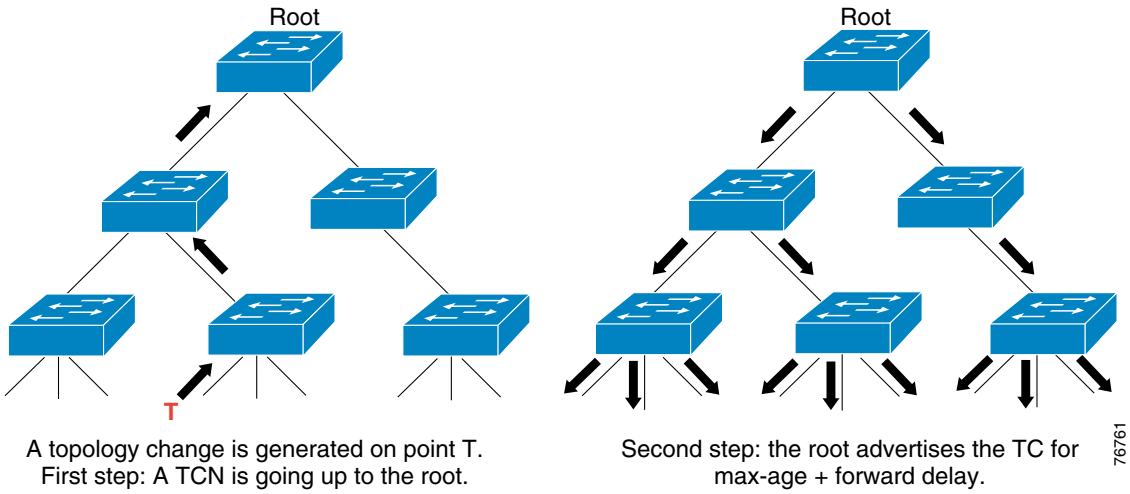
With the proposal/agreement sequence:

- The proposal agreement mechanism is very fast, as it does not rely on any timers. This wave of handshakes propagates quickly towards the edge of the network, and quickly restores connectivity after a change in the topology.
- If a designated discarding port does not receive an agreement after having sent a proposal, it slowly transitions to the forwarding state, falling back to the traditional 802.1D listening-learning sequence. This could happen for instance if the remote bridge doesn't understand RSTP BPDUs, or if the remote bridge's port is blocking.

New Topology Change Mechanisms

When an 802.1D bridge detects a topology change, it first notifies the root bridge, using a reliable mechanism, as shown in Figure 2-15.

Figure 2-15 Topology Change Mechanisms



Once the root bridge is aware of a change in the topology of the network, it sets the TC flag on the BPDUs it sends out, which are then relayed to all the bridges in the network. When a bridge receives a BPDU with the TC flag bit set, it reduces its bridging-table aging time to forward delay seconds, ensuring a relatively quick flushing of stale information. This topology change mechanism has been deeply remodeled in RSTP. Both the detection of a topology change and its propagation through the network have evolved.

Topology Change Detection

In RSTP, only non-edge ports that are moving to the forwarding state can cause a topology change. For example, a port moving to blocking no longer generates a TC. This means that a loss of connectivity is no longer considered a topology change, as was the case with 802.1D. When an RSTP bridge detects a topology change, the following happens:

- It starts the TC While timer with a value equal to twice the hello time for all its non-edge designated ports and its root port if necessary.
- It flushes the MAC addresses associated with all these ports.
- As long as the TC While timer is running on a port, the BPDUs sent out of that port have the TC bit set. BPDUs are also sent on the root port while the timer is active.

Topology Change Propagation

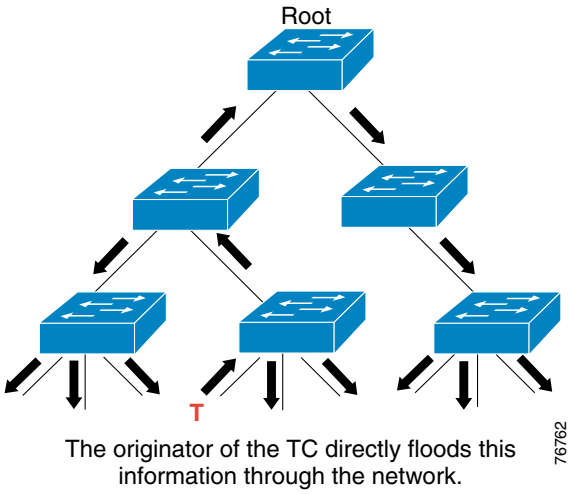
When a bridge receives a BPDU with the TC bit set from a neighbor, the following happens:

- It clears the MAC addresses learnt on all its ports except the one that received the topology change.
- It starts the TC While timer and sends BPDUs with TC set on all its designated ports and root port (RSTP no longer uses the specific TCN BPDU, unless a legacy bridge needs to be notified).

This way, the TCN is flooded very quickly across the whole network. The TC propagation is now a one step process. In fact, the initiator of the topology change is flooding this information throughout the network (as opposed to 802.1D where only the root could do so). This mechanism is much faster than

the 802.1D equivalent. There is no need to wait for the root bridge to be notified and then maintain the topology change state for the whole network for the time specified by the **max age plus forward delay** parameter (as shown in Figure 2-16).

Figure 2-16 Topology Change Propagation



In just a few seconds (a small multiple of hello times), most of the entries in the CAM tables of the entire network (VLAN) are flushed. This approach results in potentially more temporary flooding, but on the other hand it clears potential stale information that prevents rapid connectivity restitution.

Compatibility with 802.1D

As part of the RSTP standard, the RSTP BPDU format is used only when there are only RSTP speaking bridges present. 802.1D bridges do not understand the RSTP BPDUs of protocol version 2 or the MSTP BPDUs of protocol version 3 and drop them. However, RSTP bridges understand the 802.1D BPDUs of protocol version 0 for backward compatibility. If an 802.1D bridge is detected on a particular segment, an RSTP bridge will revert to using 802.1D BPDUs and TCNs on the associated port. In this way, RSTP is able to interoperate with legacy STP protocols.

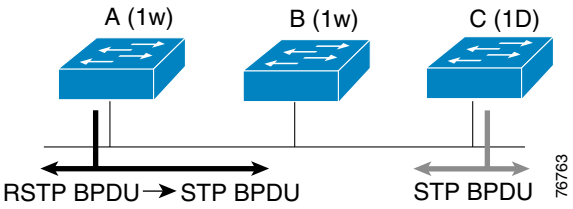
Note This means a loss of RSTP capabilities, such as rapid transition to forwarding on this segment.

Each port maintains a variable defining the protocol to run on the corresponding segment. An RSTP bridge immediately adapts the BPDUs it sends out on a port to the BPDU version it receives. A migration delay timer of twice the hello time is also started when the port comes up. While this timer is running, the current (STP or RSTP) mode associated to the port is locked. As soon as the migration delay has expired, the port will adapt to the mode corresponding to the next BPDU it receives. Thus, when an RSTP port comes up on a link where it receives 802.1D BPDUs, it waits 4 seconds before it starts sending 802.1D BPDUs.

Note If the port changes its operating mode as a result of receiving a BPDU, the migration delay is restarted, limiting the possible mode change frequency.

For example, in Figure 2-17 Bridge A and B are running RSTP with Bridge A being designated for the segment. A legacy STP Bridge C is introduced on this link.

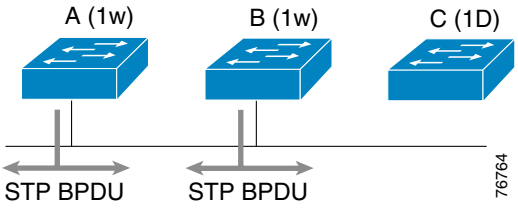
Figure 2-17 Compatibility with 802.1D—Mixture of STP Types



As 802.1D bridges ignore RSTP BPDUs and drop them, C believes there are no other bridges on the segment and starts sending its 802.1D-format BPDUs. Bridge A receives these BPDUs and, after the migration delay seconds, Bridge A changes its mode to 802.1D on that port only. As a result, Bridge C can now understand Bridge A's BPDUs and accepts Bridge A as the designated bridge for that segment, if the bridge priority for Bridge A is higher.

Note When a port is in 802.1D compatibility mode, it is also able to handle topology change notification (TCN) BPDUs, and BPDUs with TC or TCA bit set.

Figure 2-18 Compatibility with 802.1D—802.1D Bridge Drops Off



If Bridge C is removed, Bridge A will keep running in STP mode on that port even though it is able to work more efficiently in RSTP mode with its neighbor Bridge B. That is because Bridge A has no way of knowing that Bridge C has been removed from the segment. Protocol migration must then be restarted manually on the designated bridge (Bridge A) of the segment.

```
Cisco-IOS#clear spanning-tree detected-protocols interface interface_name interface_port

CatOS (enable) set spantree mst mod/port redetect-protocol
Spanning tree protocol detection forced on port mod/port.
```

For Rapid-PVST+ mode:

```
CatOS (enable) clear spantree detected-protocols mod/port
```

Note In today's networks this situation is rare because most switch-to-switch links are point-to-point direct. This situation could arise if the segment was on a full-duplex hub connecting three switches.

CHAPTER 3

Understanding Multiple Spanning-Tree Protocol (802.1s)

The Multiple Spanning Tree Protocol (MSTP) is a new standard inspired from Cisco's proprietary Multiple Instances Spanning Tree Protocol (MISTP) implementation.

Note As the number of VLANs configured in switched networks increased, Cisco developed the concept of a Multi Instance Spanning Tree Protocol (MISTP). This concept maps a number of VLANs to the same spanning tree instance by appending a list of the VLANs associated with a particular spanning tree instance. This is a relatively simple implementation that allows the number of spanning tree instances to be substantially reduced, thereby reducing system overhead associated with running STP on multiple VLANs and conserving bandwidth.

MSTP (IEEE 802.1s) standardizes the concept of multiple spanning trees, incorporating the rapid convergence offered by RSTP. MSTP allows a group of VLANs to share a spanning tree instance, defines a protocol for inter-connecting MST regions, and interoperate with existing 802.1D and 802.1q spanning tree implementation.

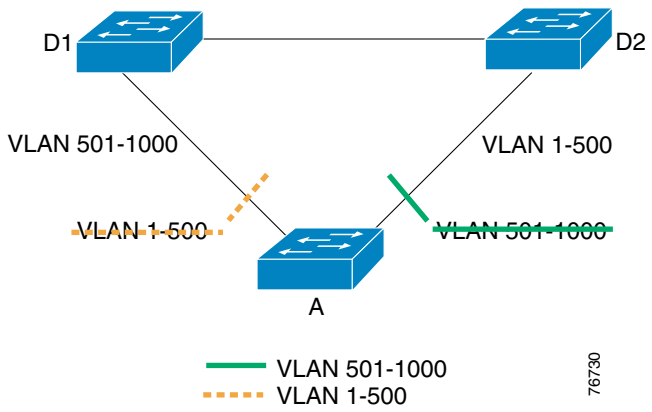
This chapter provides a comparison of MST to other forms of the STP and briefly describes the characteristics of MSTP.

Comparing MSTP with Other STPs

Figure 3-1 shows a common design featuring access Switch A with 1000 VLANs redundantly connected to distribution Switches D1 and D2. In this example, users connect to Switch A and the network administrator typically seeks to achieve load balancing on the access switch uplinks by dividing the VLANs based on a scheme deemed appropriate.

- Switch D1 is configured to be the root switch for VLANs 501 through 1000 and Switch D2 is configured to be the root switch for VLANs 1 through 500.
- The interface from Switch A to Switch D1 blocks VLANs 1 through 500 and the interface from Switch A to Switch D2 blocks VLANs 501 through 1000.

Figure 3-1 Typical VLAN Design



Per-VLAN Spanning Tree+

In a Cisco Per-VLAN Spanning Tree (PVST+) environment, one spanning-tree instance for each VLAN is maintained, which means 1000 instances for two different final logical topologies. This wastes considerable CPU cycles for all the switches in the network in (addition to the bandwidth used by each instance sending its own BPDUs).

Rapid Per-VLAN Spanning Tree+

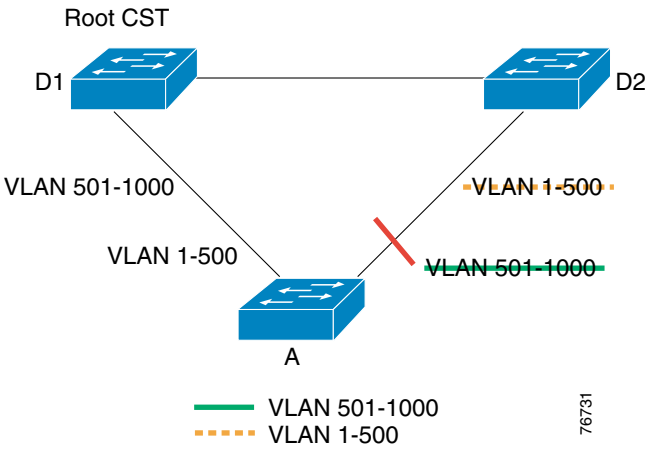
Rapid-Per-VLAN Spanning Tree (Rapid-PVST+), which is currently available only on the Catalyst 6000 (Cisco IOS and Catalyst OS) , Catalyst 3550 (Cisco IOS), and Catalyst 4000 (Catalyst OS), does not require any MST configuration. Therefore, when using limited number of VLANs, Rapid PVST+ is recommended because it gives the benefit of RSTP on a per-VLAN basis without having to configure MST.

Note For information on implementing Rapid-PVST+, see the "Configuring Rapid-PVST+" section on page 4-7.

Standard 802.1q

The original IEEE 802.1q standard defines a Common Spanning Tree (CST) that assumes only one spanning-tree instance for the entire bridged network, regardless of the number of VLANs, as shown in Figure 3-2.

Figure 3-2 Standard 802.1q Design



With CST, the following are true:

- Load balancing is not possible; one uplink needs to block for all VLANs.
- The CPU is spared; only one instance needs to be computed.

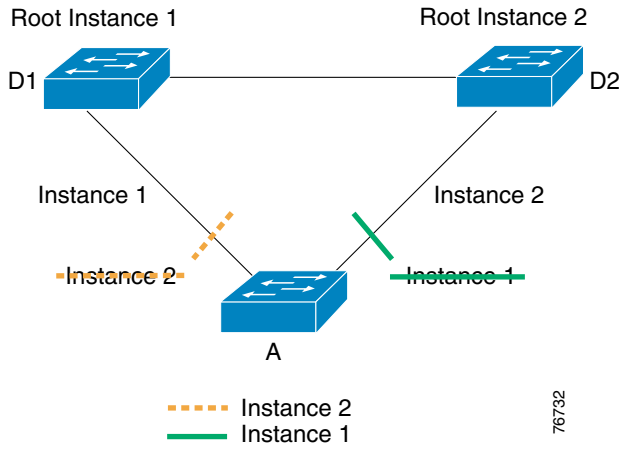
Note Cisco's implementation enhances the 802.1q in order to support PVST+. This feature behaves exactly as the PVST+ in the example above. Cisco's per-VLAN BPDUs are tunneled by pure 802.1q bridges.

Multiple Spanning Tree

Note If you are not implementing MSTP in your network, please skip to Chapter 4, "Deploying RSTP and MSTP."

MSTP (802.1s) combines the best aspects from both PVST+ and 802.1q. With MSTP, several VLANs can be mapped to a reduced number of spanning-tree instances because most networks do not need more than a few logical topologies. In the topology shown in Figure 3-2, there are only two logical topologies, so only two spanning-tree instances are necessary. If half of the 1000 VLANs are mapped to a one spanning-tree instance and other half are mapped to the other spanning-tree instance, the topology shown in Figure 3-3 results.

Figure 3-3 MSTP Design



With MSTP, the following is true:

- The desired load-balancing scheme can still be achieved because half of the VLANs follow a separate instance.
- The CPU is spared because it computes only two instances.

From a technical stand point, MSTP is the best solution. From an end-user's perspective, the only drawbacks associated with migrating to MSTP are attributed to the fact that MSTP is a new protocol. As such, the following issues arise:

- The protocol is more complex than the usual STP and requires additional training of the staff.
- Interaction with legacy bridges is sometimes challenging. See the “Interaction Between the MST Region and the Outside World” section.

MST Regions

As previously mentioned, the main enhancement introduced by MSTP is that several VLANs can be mapped to a single spanning-tree instance.The challenge is to indicate which VLAN is to be associated with which STP instance. More precisely, the question is how to tag BPDUs so that receiving devices can identify the STP instances and the VLANs to which they apply. With 802.1q, this issue is irrelevant because all instances are mapped to a unique and common instance. In the PVST+ implementation, however, different VLANs carry the BPDUs of their respective STP instance (one BPDU per VLAN).

Cisco's MISTP solved this problem by sending a BPDU for each instance that included a list of VLANs that for which the instance was responsible. However, if two switches were misconfigured and had a different range of VLANs associated to the same instance, it was difficult for the protocol to recover properly.

The IEEE 802.1s committee adopted a much easier and simpler approach by introducing *MST regions*. The association between VLANs and instances is defined in the MST region configuration. A region for spanning-tree is defined by an alphanumeric identifier, a revision number, and a table that maps the VLANs to their respective instance. The region information in the switches of that are part of the same region must match; otherwise, they will belong to different regions. The purpose of the region concept is to ensure consistent mapping between VLANs and MST instances.

MSTP Configuration and MST Region

Each switch in the network that runs MSTP has a single MSTP configuration that consists of the following attributes:

- An alphanumeric configuration name (32 bytes).
- A configuration revision number (two bytes).
- A 4096-element table that associates each of the potential 4096 VLANs supported on the chassis to a given instance.

To be part of a common MST region, a group of switches must share the same configuration attributes. It is up to the network administrator to properly propagate the configuration throughout the region. Currently, this step is only possible by the means of the Command Line Interface (CLI) or through Simple Network Management Protocol (SNMP). Other methods can be envisioned, as the IEEE specification does not explicitly mention how to accomplish this step.

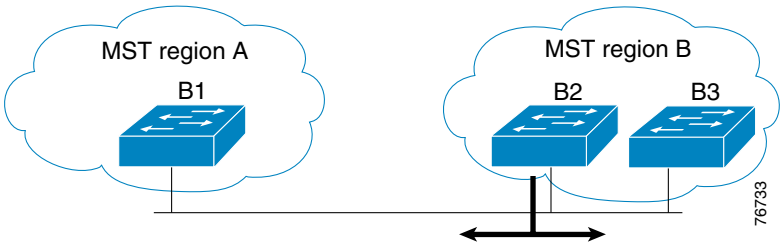
Note If two switches differ on one or more configuration attributes, then they are part of different regions.

Region Boundary

To ensure a consistent VLAN-to-instance mapping, the spanning-tree protocol must be able to identify the boundaries of the regions. For that purpose, the characteristics of the region are included in the BPDUs. The switches need only to know whether they are in the same region as their neighbor. Therefore, the exact VLANs-to-instance mapping is not propagated in the BPDUs. Instead, a digest of the VLANs-to-instance mapping table (a numerical value derived from the VLAN-to-instance mapping table through a mathematical function) is sent, along with the revision number and the configuration name. Once a switch receives a BPDU, it extracts the digest and compares it with its own computed digest. If the digests differ, the port on which the BPDU was received is the boundary of a region.

In generic terms, a port is at the boundary of a region if the designated bridge on its segment is in a different region or if it receives legacy 802.1D BPDUs. In Figure 3-4, the port on B1 is at the boundary of region A, whereas the ports on B2 and B3 are internal to region B.

Figure 3-4 Region Boundary



MST Instances

According to the IEEE 802.1s specification, an MSTP bridge must be able to handle at least the following two instances:

- One or more Multiple Spanning-Tree Instances (MSTIs)
- One Internal Spanning-Tree (IST) instance

Cisco's implementation supports 16 instances per switch: one IST (instance 0) and 15 MSTIs.

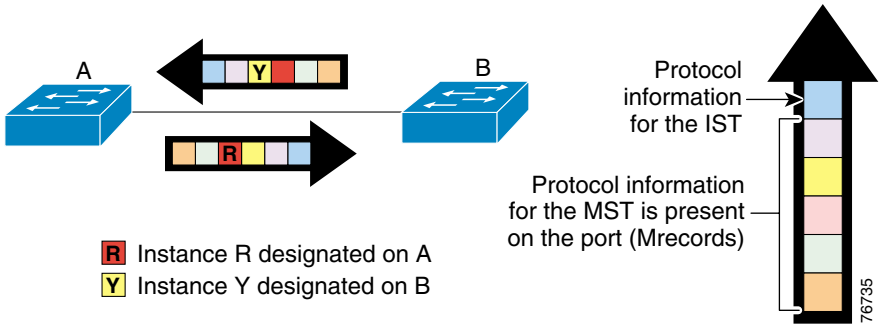
MSTIs

The MSTIs are RSTP instances that exist only within an MST region. The main difference between MSTIs and the IST is that only the IST is extended outside of the MST region. MSTIs have no interaction outside of the region and do not send BPDUs outside a region, only the IST does. Another important characteristic of MSTIs is that they do not send individual BPDUs within the MST region. Instead, bridges exchange MST BPDUs that are normal RSTP BPDUs for the IST except that they contain additional information for each MSTI.

The IST forms a single spanning tree that includes all bridges for the MST region. An MSTI is simply an IST BPDU that has additional fields appended to it, which are referred to as M-Records. It should be noted that an M-Record is only appended if the VLAN is configured on a particular port to prevent black holing of traffic.

Figure 3-5 shows two switches (A and B) exchanging BPDUs inside an MST region. Each switch sends a single BPDU and each BPDU includes an M-Record for each MSTI present on the ports.

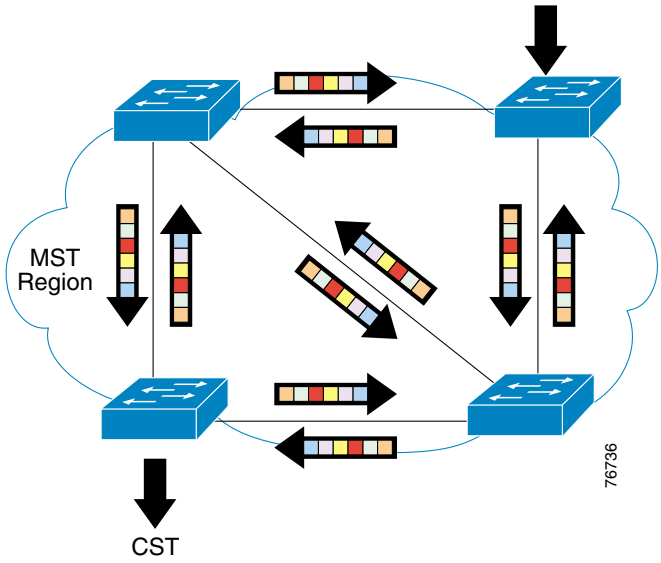
Figure 3-5 BPDUs Inside an MST Region



Note In Figure 3-5, notice that the first information field carried by an MSTP BPDU contains data about the IST. This implies that the IST (instance 0) is always present everywhere inside an MST region. However, the network administrator does not have to map VLANs onto instance 0 and, therefore, it is not a source of concern.

With a regular spanning tree, BPDUs are transmitted on the designated ports away from the root bridge. The new STP allows each switch to send and receive BPDUs simultaneously. This is because, as shown in Figure 3-6, each bridge may be designated for one or more instances, and thus needs to transmit BPDUs. As soon as a single MST instance is designated on a port, a BPDU containing the information for all instances (IST+ MSTIs) is to be sent.

Figure 3-6 BPDUs Inside and Outside an MST Region



The M-Record is a sub-field within the BPDU that contains enough information (root bridge and sender bridge priority parameters) for the corresponding instance to calculate its final topology. It does not contain any timer-related parameters (such as hello time, forward delay, and max_age) that are typically found in a regular IEEE 802.1D BPDU, as these timers are derived from the IST BPDU timers. It is important to note that within an MST region, all spanning tree instances use the same parameters as the IST.

The hello parameter is always relevant as it is used to determine the frequency that BPDUs are transmitted between switches. Inside a region, the forward delay parameter is used when two bridges are communicating via a shared link (such as a half-duplex connection), or when an RSTP BPDU is unacknowledged on a port that wants to transition to forwarding. When interacting with the outside world (such as with a PVST bridge running 802.1D or a different region), the IST uses all of the timer-based parameters.

IST

MSTP originates from the IEEE. Therefore, MSTP must be able to interact with 802.1q-based networks. For 802.1q, a bridged network implements only a single spanning-tree (CST).

The IST (instance 0) runs on all bridges within an MST region. An important characteristic of the IST instance is that it provides interaction at the boundary of the MST region with other MST regions and, more importantly, it is responsible for providing compatibility between the MST regions and the spanning tree of 802.1D, 802.1q (CST) and PVST+ networks connected to the region.

RSTP is not a timer-based protocol. But for the IST to be backwardly compatible and for it to communicate with devices outside the region, it needs to carry the timers. Therefore, the IST carries legacy STP timers, such as max-age, fwd-delay, and hello.

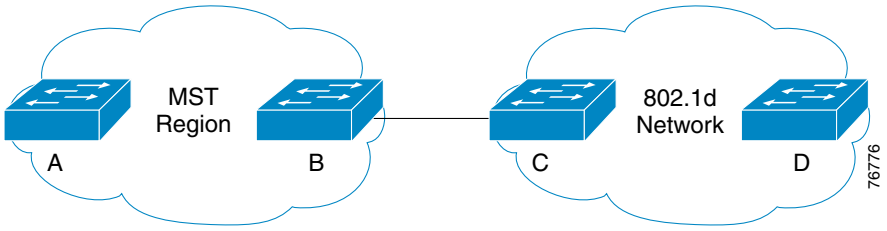
- RSTP slowly transitions a shared link to forwarding. Slowly is defined as a wait time of twice the fwd-delay in seconds. A shared link is a half-duplex link. RSTP also slowly transitiitions a link to forwarding if a proposal was sent but an agreement was not received.
- RSTP rapidly transitions a link to forwarding only if it is a point-to-point link. A point-to-point link is determined when the link is in full-duplex mode.

MST Hop Count

MST uses the concept of a hop count, which first gets assigned by the root of the spanning tree and is decremented by one through each Layer 2 hop as the BPDU traverses through the region. If the hop count reaches 0, then the received BPDU is discarded. The default hop count is 20, which means the farthest a bridge can be from its root is 20 hops. This should not be a limitation in a campus network. Cisco's recommended designs do not place switches more than 2-3 Layer 2 hops away from the root.

In Figure 3-7, the root for the MST region is Bridge B. That means that there can be 20 Layer 2 hops from A to B. Because Bridge B is the boundary of MST region, IST there interacts with Bridge C. The MST hop count has no meaning to 802.1D. The incoming BPDUs at Bridge C are in the 802.1D format. Assume that root for the 802.1D region is inside the MST region, as per the recommendation later. 802.1D Configuration BPDUs leaving Bridge B toward Bridge C will have a Message-Age of 0 and from Bridge C on its journey toward Bridge D. The Message-Age is incremented by 1 by every bridge hop in the 802.1D network. This is normal 802.1D operation after the BPDU leaves the root (in this case Bridge B). Therefore, BPDUs leaving Bridge B are of 802.1D format because Bridge C only understands 802.1D.

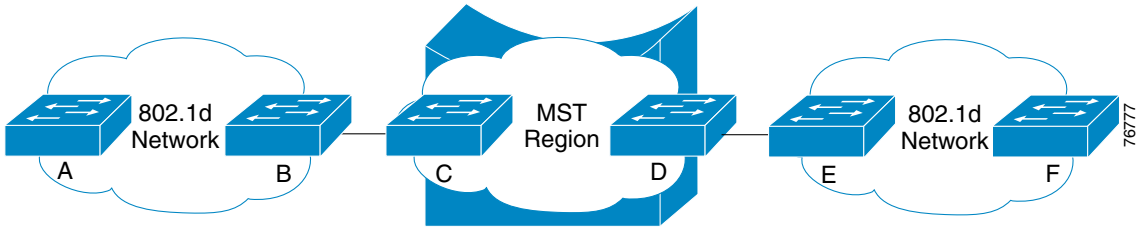
Figure 3-7 MST Region Interaction with 802.1D Network



Theoretically, the MST root can be in the middle between Bridge A and Bridge B. That results in 40 hops between Bridge A and Bridge B because the MST hop count limitation is counted as 20 hops from the root. However, this is not practical because failure of the root switch in middle would immediately produce a hop count greater than 20 for the switch that is farthest away from the root.

In Figure 3-8, Bridge C and Bridge D are at the boundary of the MST region and is, therefore, interacting with legacy 802.1D on both their ends and speaking 802.1D to be compatible with Bridge B and Bridge E. The MST region in the middle appears as one big 802.1D switch to the 802.1D networks on both sides. According to 802.1D, the recommended maximum number of bridge hops between any two end stations is seven. Therefore, there can be seven Layer 2 hops between Bridge A and Bridge F. The whole MST region in the middle counts as one switch hop irrespective of how many switches are inside the region.

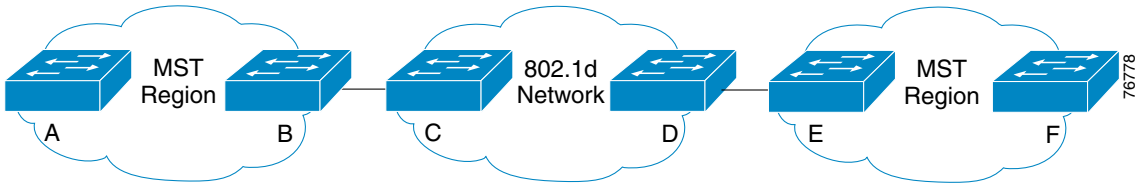
Figure 3-8 MST Region in Middle of Two 802.1D Networks



In Figure 3-9, legacy 802.1D region will cause the two MST regions to split because 802.1D does not understand MST BPDUs. And there is no tunneling mechanism between Bridge B and Bridge E. As before, Bridge B and Bridge E are at the boundary interacting with Bridge C and Bridge D. The journey

from Bridge A to Bridge D is similar to that shown in Figure 3-7. If the MST root is at Bridge B and Bridge E for the two regions then, there are 20 hops each from Bridge A to Bridge B and from Bridge F to Bridge E. Between Bridge B and Bridge E is the 802.1D region and its limitation is seven hops.

Figure 3-9 802.1D Network in the Middle of Two MST Regions

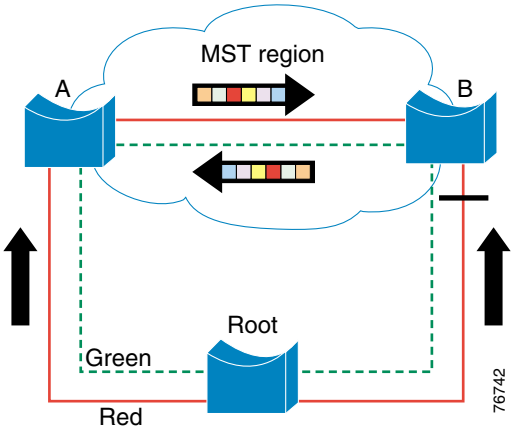


Interaction Between the MST Region and the Outside World

When migrating to an MSTP network, the administrator is likely to have to deal with interoperability issues between MSTP and legacy protocols. MSTP seamlessly interoperates with standard 802.1q CST networks, however, only a handful of networks are based on the 802.1q standard because of its single spanning-tree restriction. Cisco provides an efficient yet simple compatibility mechanism between MSTP and PVST+.

The first property of an MST region is that at the boundary ports no MSTI BPDUs are sent out, only IST BPDUs are sent. Internal instances (MSTIs) always automatically follow the IST topology at boundary ports, as shown in Figure 3-10.

Figure 3-10 MSTP at Boundary Ports

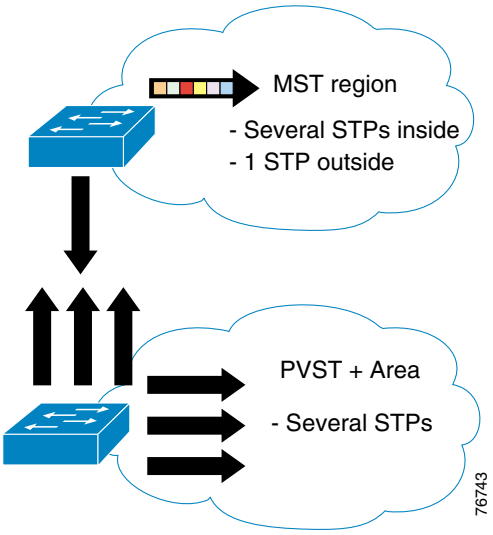


Assume VLANs 10-50 are mapped to the green instance, which is an internal instance (MSTI) only. The red links represent the IST, and therefore they also represent the CST. VLANs 10-50 are allowed everywhere in the topology. BPDUs for the green instance are not sent out of the MST region. This does not mean that there is a loop in VLANs 10-50. MSTIs follow the IST at the boundary ports, and the boundary port on Bridge B will also block traffic for the green instance.

Switches running MSTP are able to automatically detect PVST+ neighbors at boundaries. They do so by detecting that multiple BPDUs are received on different VLANs of a trunk port for the instance.

Figure 3-11 shows an interoperability issue. An MST region only interacts with one spanning-tree (CST) outside of the region. However, PVST+ bridges run one STA per VLAN and, as a result, send one BPDU on each VLAN every two seconds. The boundary MSTP bridge does not expect to receive that many BPDUs. It expects to either receive one or to send one, depending on whether it is the root of the CST or not.

Figure 3-11 MSTP Interoperability Issue

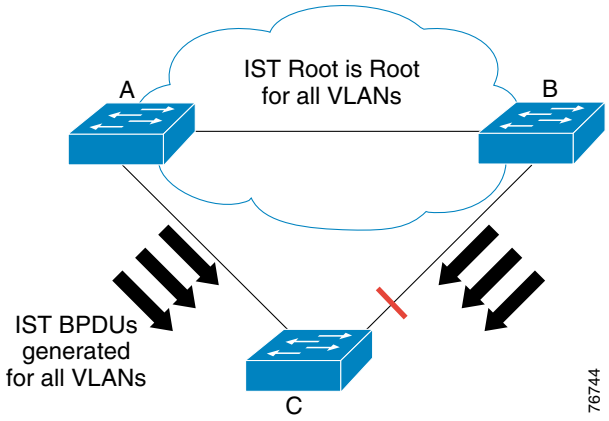


Cisco developed a mechanism to address the problem shown in Figure 3-11. A possibility would have consisted in tunneling the extra BPDUs sent by PVST+ bridges across the MST region. However, this solution has proven to be too complex and potentially dangerous when first implemented in the MISTP. A simpler approach was created. The MST region simulates a PVST+ neighbor by replicating the IST BPDU on all the VLANs. This solution implies a few constraints that are discussed in the following section.

Recommended Configuration

As the MST region now replicates the IST BPDUs on every VLAN at the boundary, each PVST+ instance will hear a BPDU from the IST root (this implies the root is located inside the MST region). Cisco recommends that the IST root have a better priority than any other bridge in the network so that the IST root becomes the root for all the different PVST+ instances, as shown in Figure 3-12.

Figure 3-12 IST as Root—Recommended Configuration



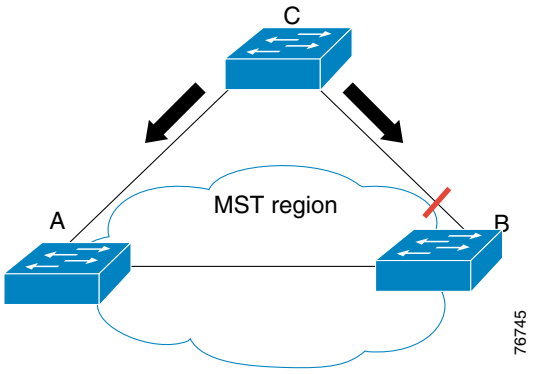
In Figure 3-12, Bridge C is a PVST+ redundantly connected to an MST region. The IST root is the root for all PVST+ instances existing on Bridge C. As a result, Bridge C blocks one of its uplinks in order to prevent loops. In this particular case, interaction between PVST+ and the MST region is optimal for the following reasons:

- Bridge C's uplink ports costs can be tuned to achieve load balancing of the different VLANs across the uplinks' ports (because Bridge C runs one spanning-tree per VLAN, it is able to chose which uplink port will block on a per-VLAN basis).
- Uplink fast can be used on Bridge C to achieve fast convergence in case of an uplink failure.

Alternate Configuration (Not Recommended)

Another possibility is to have the IST region be the root for absolutely no PVST+ instance. This means that all PVST+ instances have a better root than the IST instance, as shown in Figure 3-13.

Figure 3-13 IST not Root—Not Recommended



This case corresponds to a PVST+ core and an MSTP access or distribution layer, a rather infrequent scenario. Establishing the root bridge outside the region has the following drawbacks compared to the previously recommended configuration:

- An MST region only runs one spanning-tree instance that interacts with the outside world. This means that a boundary port can only be blocking or forwarding for all VLANs. There is no load-balancing possible between the region's two uplinks leading to Bridge C. The uplink on Bridge B, for the instance, will be blocking for all VLANs while Bridge A will be forwarding for all VLANs.
- This configuration still allows for fast convergence inside the region. If the uplink on Bridge A fails, a fast switch over to an uplink on a different switch needs to be achieved. While the way the IST behaves inside the region in order to have the whole MST region resemble a CST bridge was not discussed in details, you can imagine that a switchover across a region will never be as efficient as a switchover on a single bridge.

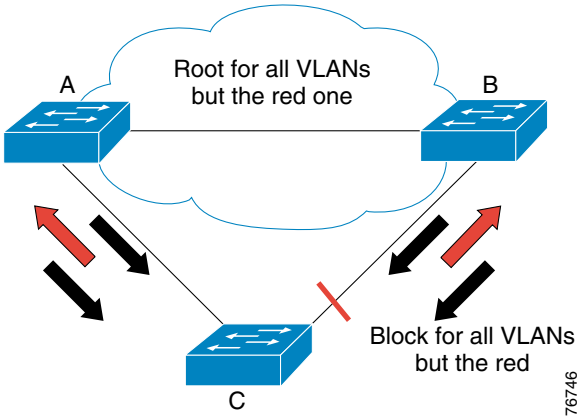
Invalid Configuration

While providing easy and seamless interoperability between MSTP and PVST+, the PVST+ emulation mechanism implies that any configuration other than the two previously mentioned ones is invalid. The following are the basic rules that must be followed to get a successful MSTP/PVST+ interaction:

- If the MSTP bridge is the root, it must be the root for all VLANs.
- If the PVST+ bridge is the root, it must be the root for all VLANs (including the CST, which always runs on VLAN 1, regardless of the native VLAN, when running PVST+).
- The simulation will fail and produce an error message if the MSTP bridge is the root for the CST, while the PVST+ bridge is the root for one or more other VLANs. A failed simulation puts the boundary port in root inconsistent mode.

In Figure 3-14, Bridge A in the MST region is the root for all three PVST+ instances except one (the red VLAN). Bridge C is the root of the red VLAN. Suppose that the loop created on the red VLAN, where Bridge C is the root, becomes blocked by Bridge B. This would mean that Bridge B would be designated for all VLANs except the red one. An MST region is not able to do that. A boundary port can only be blocking or forwarding for all VLANs because the MST region is only running one spanning-tree with the outside world. Thus, when Bridge B detects a better BPDU on its boundary port, it invokes the BPDU guard to block this port. The port is placed in the root inconsistent mode. The exact same mechanism will also lead Bridge A to block its boundary port. Connectivity is lost, however, a loop-free topology is preserved even in the presence of such a misconfiguration.

Figure 3-14 Invalid Configuration



Note As soon as a boundary port produces a root inconsistent error, start investigating whether a PVST+ bridge is attempting to become the root for some VLANs.

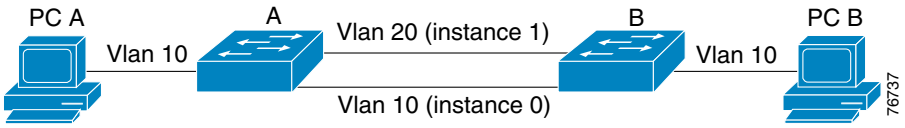
Common Misconfigurations

The independence between instance and VLAN is a new concept that implies careful configuration planning. This section illustrates some common pitfalls and how to avoid them.

IST Instance is Active on All Ports, Whether Trunk or Access

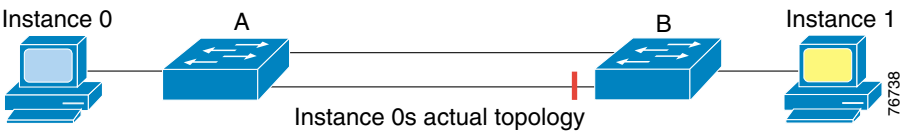
Figure 3-15 shows Bridge A and Bridge B connected using access ports located in different VLANs. VLAN 10 and VLAN 20 are mapped to different instances. VLAN 10 is mapped to instance 0 and VLAN 20 is mapped to instance 1.

Figure 3-15 Two VLANs, Two Instances



This configuration results in PC A not being able to send data to PC B. Issuing the show command reveals that Bridge B is blocking its link to Bridge A in VLAN 10, as shown in Figure 3-16.

Figure 3-16 Interruption in Service

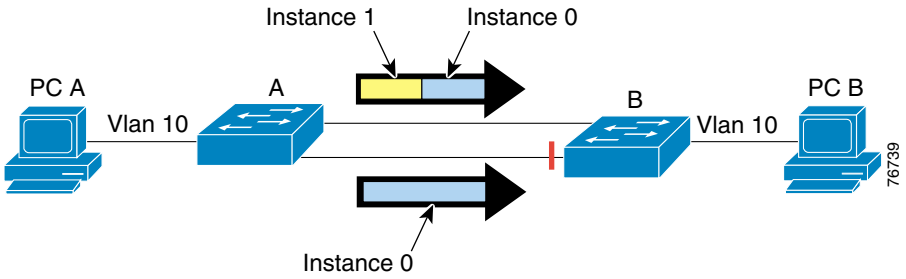


How is that possible in such a simple topology, with no apparent loop?

This issue is explained by the fact that MSTP information is conveyed using only one BPDU (an IST BPDU), regardless of the number of internal instances. Individual instances do not send individual BPDUs. When Bridge A and Bridge B exchange STP information for VLAN 20, they send an IST BPDU with an MRecord for instance 1 because that is where VLAN 20 is mapped. However, because it is an IST BPDU, it also contains information for instance 0. This means that the IST instance is active on all ports inside an MST region, whether these ports carry VLANs mapped to the IST instance or not.

Figure 3-17 shows the logical topology of the IST instance.

Figure 3-17 Logical Topology



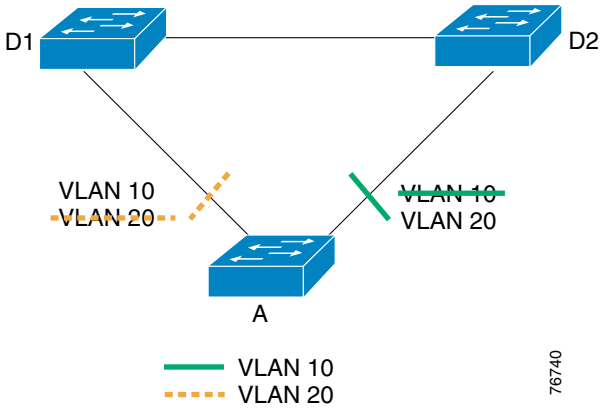
Bridge B receives two BPDUs for instance 0 from Bridge A (one on each port). It is clear that Bridge B has to block one of its ports in order to avoid a loop.

The preferred solution is to avoid mapping VLANs to the IST instance by using one instance for VLAN 10 and another instance for VLAN 20. An alternative is to carry those VLANs mapped to the IST on all links (allow VLAN 10 on both ports).

Two VLANs Mapped to the Same Instance Will Block the Same Ports

Remember that a VLAN no longer equates to a spanning-tree instance. The topology is determined by the instance, regardless of the VLANs mapped to it. Figure 3-18 shows a problem that is a variant of the one discussed in the previous section.

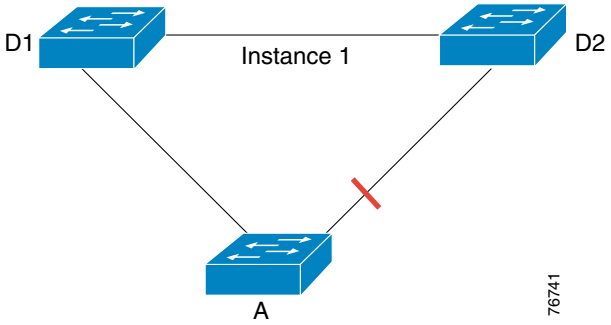
Figure 3-18 Two VLANs; One Instance



In this example, VLANs 10 and 20 are both mapped to the same instance (instance 1). The network administrator wants to restrict traffic on the uplink trunks from Bridge A to distribution Switches D1 and D2 by manually pruning VLAN 10 on one uplink and VLAN 20 on the other (trying to achieve a topology as shown in Figure 3-18). Shortly after having done that, the network administrator notices that users in VLAN 20 have lost connectivity to the network.

This is a typical misconfiguration problem. VLANs 10 and 20 are both mapped to instance 1, meaning there is only one logical topology for both VLANs. Load-sharing cannot be achieved, as shown in Figure 3-19.

Figure 3-19 Load-sharing with MSTP



Because of the manual pruning, VLAN 20 is only allowed on the blocked port, which explains the loss of connectivity. To achieve load balancing, the network administrator should have mapped VLAN 10 and 20 to two different instances.

Note A simple rule to follow to avoid this problem is to never manually prune VLANs off a trunk. If some VLANs must be removed from a trunk, then all the VLANs mapped to a given instance should be removed together. Never remove an individual VLAN from a trunk without removing all the VLANs that are mapped to the same instance.

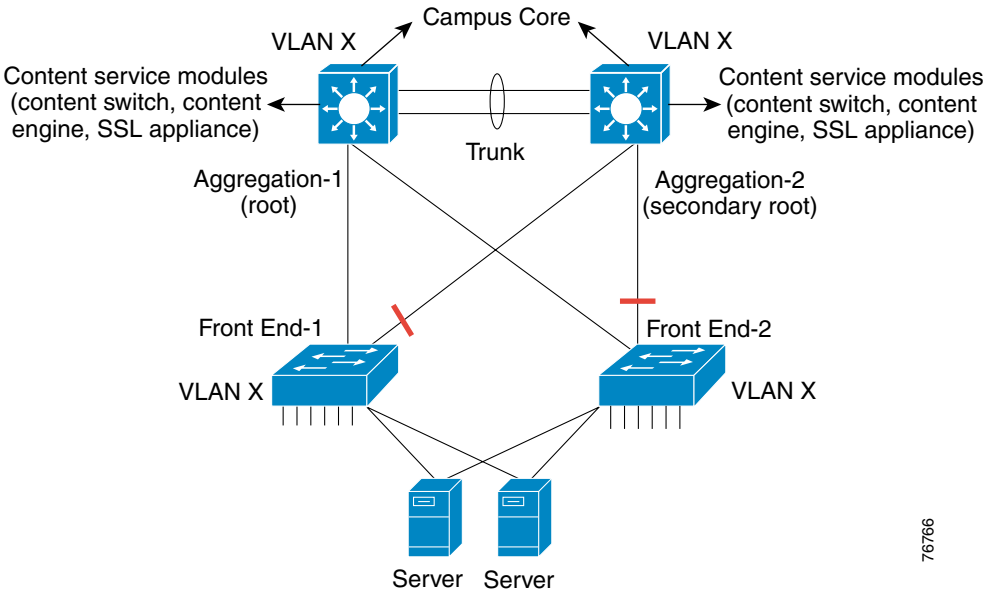
Deploying RSTP and MSTP

This chapter describes how to implement RSTP and MSTP in a campus network, particularly in the data center. It examines the topology and provides sample configurations.

Data Center Topology

Figure 4-1 represents elements of a data center in a campus, which could host various content elements and most importantly the server farm.

Figure 4-1 Campus Data Center



Cisco recommends that you have *one or two spanning-tree topologies* in the data center. In the event that load balancing of traffic to the uplinks from the access to the aggregation switches is needed, two instances should be configured in addition to the IST. The conclusion is that the total number of instances should range between 2 and 3 depending on the configuration, still a small number when compared to the number of spanning-tree instances that the switch had to maintain with PVST+.

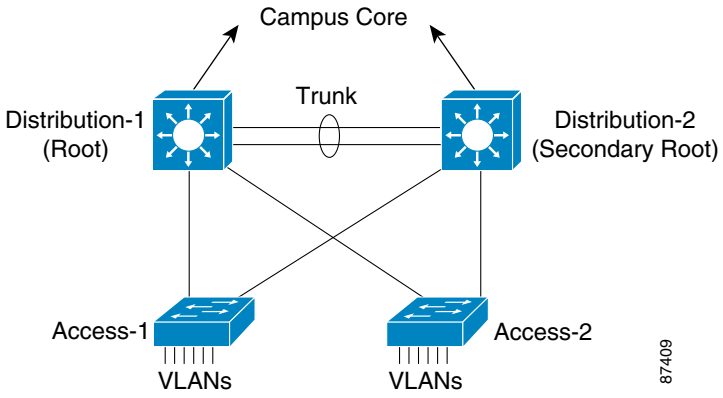
RSTP Active Topology

The spanning-tree active topology computed by RSTP is the same as that for 802.1D. However, RSTP uses a different algorithm, IEEE 802.1w. The following sections illustrate RSTP behavior.

RSTP Convergence Example

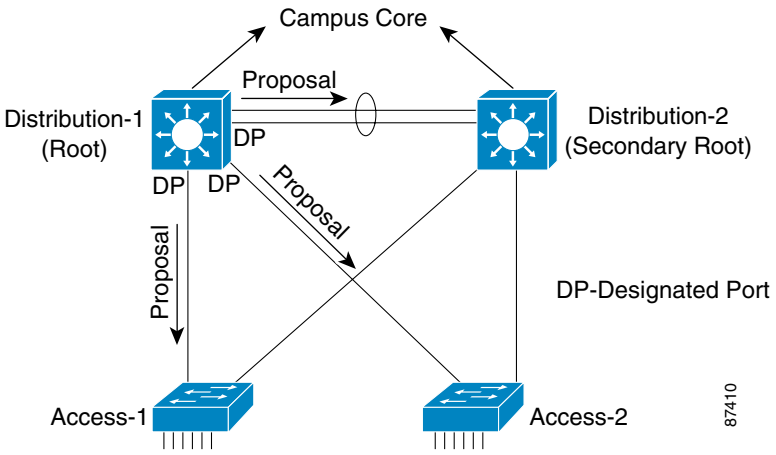
In RSTP convergence, the links are placed in designated blocking when they first come up. RSTP relies on a 2-way handshake mechanism to rapidly transition a designated port to forwarding. To illustrate, assume that in Figure 4-2 Distribution-1 has the lowest bridge priority.

Figure 4-2 RSTP Convergence—Initial State



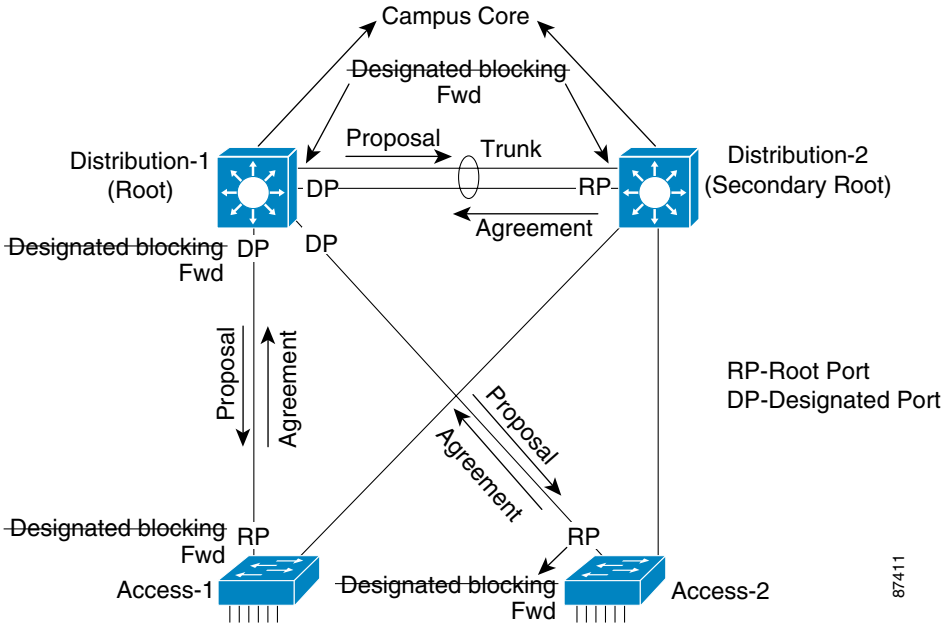
Convergence is as follows:

Step 1 Distribution-1 sends a configuration RSTP BPDU with the Proposal flag set from all its designated ports. The intent is to convey to the neighboring switches (neighboring root ports) that Distribution-1 would like to assume designated port status for the segment and that it wants to progress rapidly to forwarding.



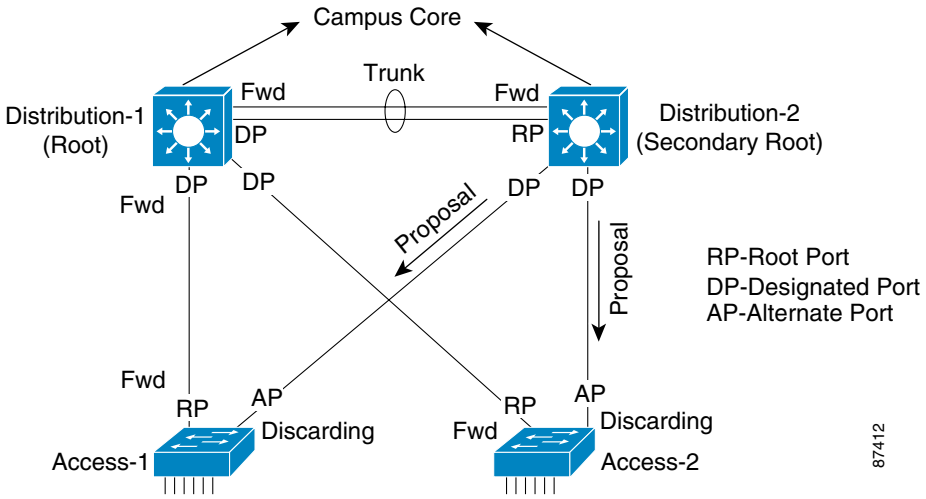
Step 2 When the neighboring switches see an RSTP BPDU with the Proposal flag set and the BPDU has a better bridge priority than their own, they stop sending their own configuration BPDUs and start to calculate their cost to the root bridge in the same manner as in 802.1D. Access-1, for example, starts a "sync" process. The sync ensures that there are no possible loops past Access-1 by reverting the designated ports of Access-1 (not depicted in Figure 4-2) to discarding (blocking).

Step 3 After all the ports on Access-1 are synced, the root port of Access-1 transmits the proposal RST BPDU back to Distribution-1 with the agreement flag set. At the same time, Distribution-2 and Access-2 also complete a similar handshake with Distribution-1 to rapidly transition their links to forwarding. Upon receiving this BPDU (Agreement), Distribution-1 transitions its port to forwarding without further delay and becomes the designated port on the respective segments.

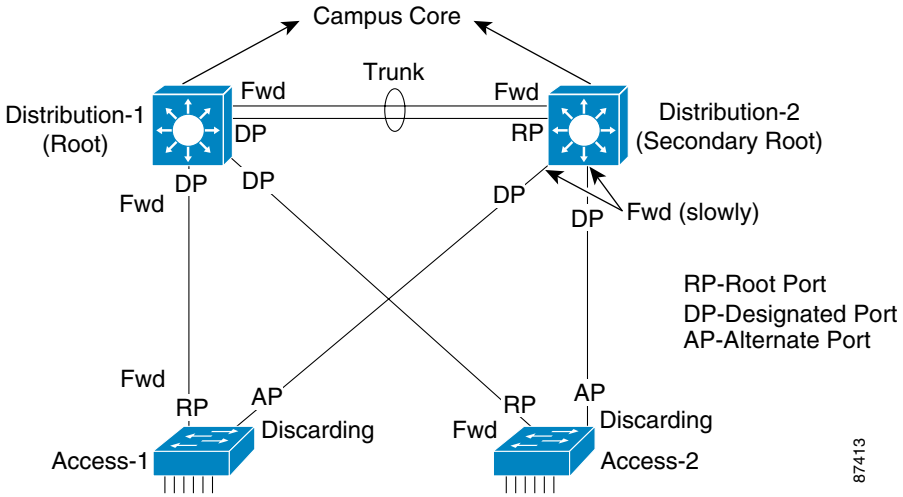


Step 4 Distribution-2 sends a configuration BPDU with the Proposal flag set to Access-1 and Access-2. This BPDU has the latest information, such as the Root Bridge ID of Distribution-1 and the Root Port Cost to reach Distribution-1. The RSTP BPDU also carries locally significant variables, such as the Designated Bridge ID and the Designated Port ID. These variables are used only as the tie breakers, if needed.

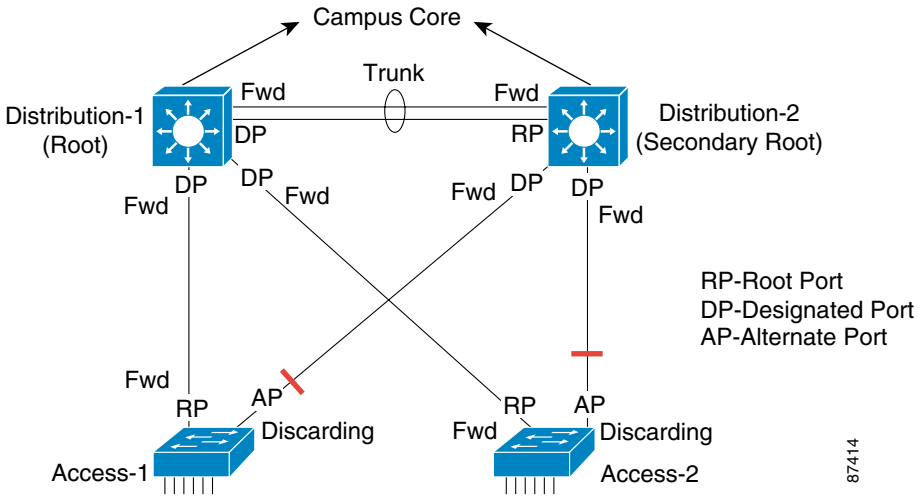
Access-1 receives the BPDU from Distribution-2 and knows immediately that the BPDU is inferior because Access-1 has already established a forwarding link with the root (Distribution-1). Therefore, Access-1 does not respond to the proposal and considers the link to Distribution-2 to be an alternate port (putting it into discarding state).



Step 5 Distribution-2 did not receive an agreement for its proposal BPDU. Therefore, it will not *rapidly* transition its links to Access-1 or Access-2 to forwarding. Distribution-2 instead *slowly* transitions its links to Access-1 and Access-2 to the forwarding state. *Slowly* is defined as a wait time of twice the fwd-delay. This is also that amount of time it takes 802.1D to transition a link to forwarding (listening plus learning, in seconds).



Step 6 In the resulting topology, Distribution-1 is the root. Therefore, Access-1 and Access-2 consider their links to Distribution-2 as alternate root ports (alternate port in 802.1w terminology) and put them in discarding state.



The side effect of the handshake is that the sync process can be viewed as a cut in the active topology because during the handshake, a bridge puts its designated ports in sync. This is the only way to guarantee fast transition. The cut is propagated from the original designated port through all segments in the sub-tree below it (in the direction away from the Root) until the cut reaches the edge of the network. The handshake is very fast because it does not rely on timers. And the wave of cuts quickly moves toward the edge, restoring connectivity after a topology change.

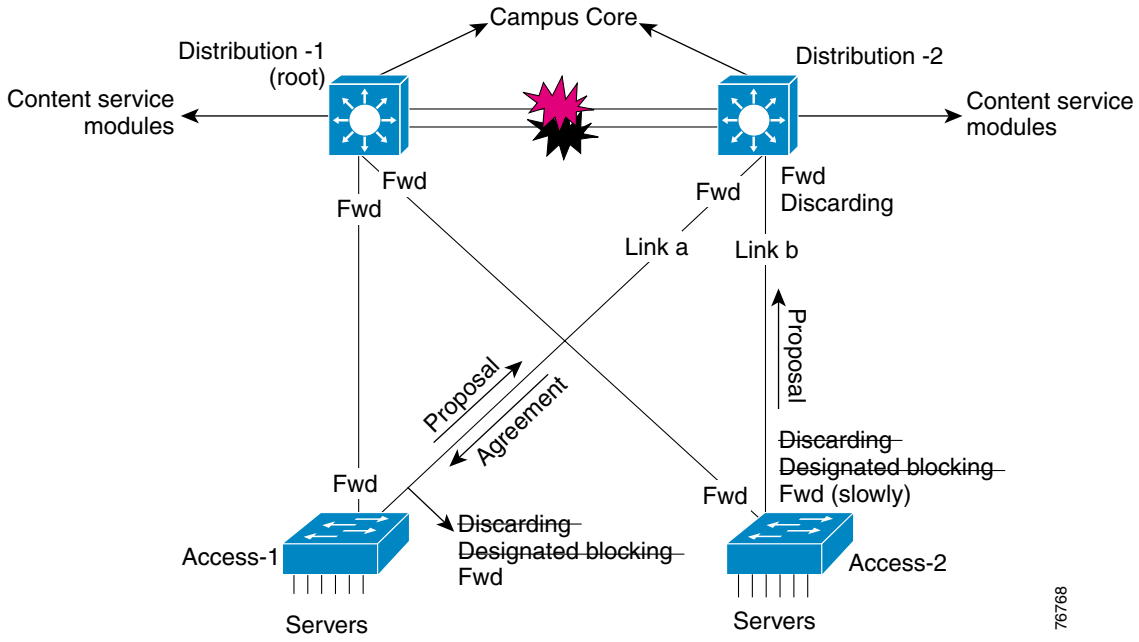
It is also important to note that the edge ports (PortFast enables ports) are not put into a discarding state during the sync process. In fact, in most cases, the Cisco implementation of the sync process in 802.1w puts only the bridge's former root port into a discarding state during a sync.

RSTP Link Failure Recovery

If the link between Distribution-1 and Distribution-2 breaks (shown in Figure 4-3), then it is considered an indirect root port failure. Cisco's extension to 802.1D introduced the BackboneFast feature, which speeds the convergence by skipping the max-age timer. However, it still takes twice the fwd-delay (30 seconds, by default) to converge after this type of a failure.

RSTP has a built-in mechanism to rapidly recover from indirect failures.

Figure 4-3 Indirect Root Link Failure



With RSTP:

Step 1 As soon as the link between the primary root (Distribution-1) and the secondary root (Distribution-2) is lost, Distribution-2 starts transmitting Configuration BPDUs advertising itself as the root.

Step 2 Access-1 and Access-2 realize that they are receiving inferior BPDUs on their alternate ports as they still have their connection with the root (Distribution-1). Access-1 and Access-2 then enter designated blocking state on their links to Distribution-2 and send a Proposal BPDU to Distribution-2 with information about the Root (Distribution-1).

Step 3 Distribution-2 receives two Proposals, one from Access-1 and the other from Access-2. It compares the two BPDUs and selects the best path to the root (as with 802.1D).

Step 4 In this example, the cost to the root for Distribution-2 is same via Access-1 or Access-2. Assuming that Access-l has a lower sender bridge ID than Access-2, then Distribution-2 sends an Agreement to Access-1 on link A (the new root port of Distribution-2) to rapidly transition to forwarding.

Step 5 Access-1 transitions to forwarding immediately.

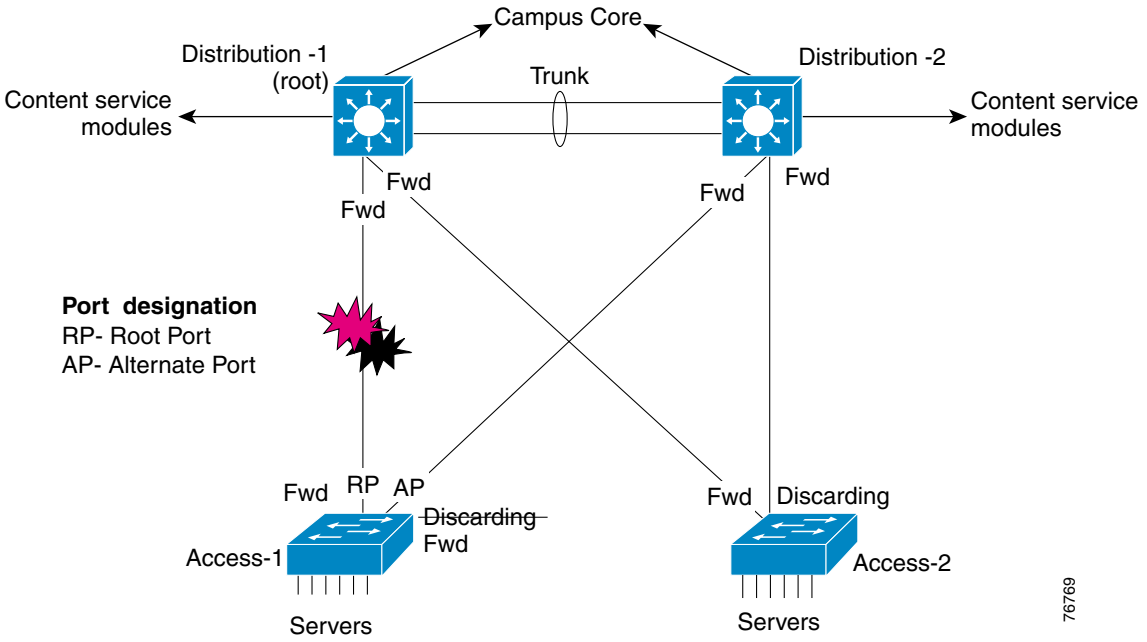
Step 6 Distribution-2 does not reply to Access-2's proposal on link B and consider it as an alternate port (putting it in discarding state).

Step 7 Access-2 never receives an agreement for its proposal. Therefore, it transitions the port to Distribution-2 to the forwarding state in 30 seconds (twice the fwd-delay).

If the link between Access-1 and Distribution-1 fails (as shown in Figure 4-4) then it is considered a direct root port failure. Cisco's extension to 802.1D introduced the UplinkFast feature, which upon detecting a direct link failure of the root port transitions an alternate blocking port directly to forwarding (assuming the root port and alternate port are on the same switch).

RSTP has a built-in mechanism to do the same without the need to configure UplinkFast on Access-1 or Access-2.

Figure 4-4 Direct Root Link Failure



After the root port failure, the switches realize that they do not have a root port on the segment and rapidly transition the alternate port from discarding to forwarding.

With UplinkFast, there is a concept of dummy multicast packets, which are sent out on the new root port to update the CAM table on the upstream switch. RSTP does not require dummy multicasts because invalid entries are immediately flushed from the CAM table as a result of the new topology change mechanism. The previous standard waited for the fwd-delay time (15 seconds, by default) to age out stale CAM entries during topology changes.

Note A link moving from discarding (blocking) to forwarding in RSTP causes the bridge to send a Topology Change Notification (TCN). Upon receiving a TCN, RSTP bridges flush their forwarding table (CAM table) for all ports except edge ports and the port that received the TCN.

Configuring Rapid-PVST+

Running RSTP on a per-VLAN basis without MST was not possible until Cisco released the Rapid-PVST+ feature. In situations where there are not many VLANs and there is no need to group multiple VLANs into an instance, Rapid-PVST+ gives the benefit of RSTP fast convergence on a per-VLAN basis. Each VLAN runs its own RSTP instance, is responsible for sending RSTP BPDUs, and maintains its own spanning tree active topology.

Rapid-PVST+ does not require any MST parameters, such as configuration name, revision number, or instance-to-VLAN mapping. Also, there is no concept of instance 0. Therefore, when using a limited number of VLANs, Rapid PVST+ is recommended.

Rapid-PVST+ uses protocol version 2 for RSTP BPDUs, unlike MSTP BPDUs, which use protocol version.

As with MSTP, Rapid-PVST+ is configured from the spanning tree global configuration mode.

Note Unlike MST, Rapid-PVST+ does not require MAC address reduction. However, there are situations, such as running 4000+ VLANs that could require the use of MAC address reduction.

Example 4-1 and Example 4-2 show the configuration for IOS. Example 4-1 and Example 4-2 show the configuration for Catalyst OS.

Example 4-1 Configuring Rapid-PVST+—IOS

```
CiscoIOS#config t
CiscoIOS(config)#spanning-tree extend system-id          Enables MAC Address reduction, also known
!                                                          as extended system-id in IOS.
CiscoIOS(config)#spanning-tree mode rapid-pvst+        Sets spanning tree mode to Rapid-PVST+.
CiscoIOS(config)#spanning-tree vlan vlan_number root primary       Makes this switch root or
CiscoIOS(config)#spanning-tree vlan vlan_number root secondary     secondary root for the
                                                                   VLANs specified.
CiscoIOS(config)#exit
```

Example 4-2 Verifying Rapid-PVST+—IOS

```
CiscoIOS#show running-config
Excerpt
spanning-tree mode rapid-pvst+
spanning-tree extend system-id
spanning-tree vlan_number priority 24576              Priority value is the result of "spanning-tree
!                                                       vlan <vlan> root primary" command.
spanning-tree vlan_number priority 28672              Priority value is the result of "spanning-tree
!                                                       vlan <vlan> root secondary" command.

CiscoIOS#show spanning-tree vlan 30                   Show commands are VLAN based rather than
!                                                       than Instance based.
VLAN0030
  Spanning tree enabled protocol rstp
  Root ID    Priority    24606
             Address     00d0.047b.2800
             This bridge is the root
             Hello Time   2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID    Priority    24606  (priority 24576 sys-id-ext 30)   MAC Address Reduction caused
!                                                               the bridge priority 24576 to
!                                                               increment by 30 because the
!                                                               VLAN number is 30.
             Address     00d0.047b.2800
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Desg FWD 4         128.1    P2p
Gi1/2            Desg FWD 4         128.2    P2p
Gi5/1            Desg FWD 4         128.257  P2p
```

Example 4-3 Configuring Rapid-PVST+—Catalyst OS

```
CatOS (enable) set spantree mode rapid-pvst+          Sets spanning tree mode to Rapid-PVST+
CatOS (enable) set spantree macreduction enable       Enables MAC Address reduction, also
!                                                       known as extended system-id in IOS.
CatOS (enable) set spantree root vlan_number
CatOS (enable) set spantree root secondary vlan_number    Makes this switch root or secondary root
!                                                           for the VLANs specified.
```

Example 4-4 Verifying Rapid-PVST+—IOS

```
CatOS (enable) show config
Excerpt
#mac address reduction
set spantree macreduction enable
!
#stp mode
set spantree mode rapid-pvst+
!
#spantree
!
set spantree priority 24576 vlan_number          Priority value is the result of "set  spantree root
!                                                <vlan>" command.
set spantree priority 28672 vlan_number          Priority value is the result of "set spantree root
!                                                secondary vlan" command.
```

Configuring MSTP

This section provides instructions and examples for configuring MSTP. Specifically, it discusses:

- MST Region
- MAC Address Reduction
- Configuring MSTP at the Distribution Level
- Configuring MSTP at the Access Layer

Note Read this section only if you are planning to implement MSTP.

Note RSTP was first implemented as part of Multiple Spanning-Tree Protocol (MSTP) in Catalyst OS 7.1 and IOS software release 12.1(11)EX and later. It is currently available as a standalone protocol with the Rapid-PVST+ mode on the Catalyst 6000 in IOS software release 12.1(13)E and Catalyst OS 7.4, on the Catalyst 3550 switch in IOS software release 12.1(13)EA1, and on the Catalyst 4000 in Catalyst OS 7.4. In this mode, the switch runs an RSTP instance on each VLAN. Rapid-PVST+ is discussed in the "Configuring Rapid-PVST+" section on page 4-7.

MST Region

All bridges within an MST region must agree on a common MSTP identifier derived from the MSTP configuration. The identifier is checked before the bridges become part of a region. The elements that make up the identifier are listed below and configured as part of the MSTP configuration:

- Configuration name
- VLAN to instance mapping
- Revision number

It is recommended that all switches running MST to be in the same region.

MAC Address Reduction

802.1D states that each bridge must have a unique bridge identifier. In PVST, each VLAN is considered as a different logical bridge. Therefore, each VLAN needs a unique bridge identifier. Prior to supporting 4000 VLANs, Cisco supported a maximum of 1024 VLANs, which required 1024 bridge identifiers.

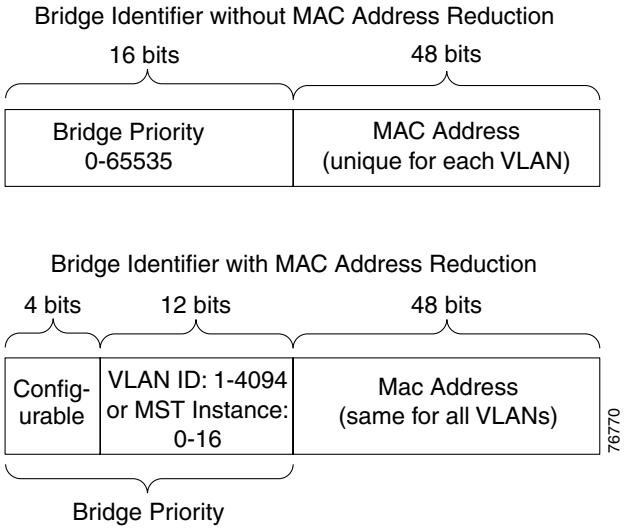
MAC address reduction is a feature that ensures bridge ID uniqueness for all 4000 VLANs even when there are only 1024 or 64 MAC addresses available on the switch. It accomplishes this by making the 16-bit Bridge Priority field in the BPDU unique for each VLAN. Prior to this feature, the Bridge Priority was fully configurable and did not have to be unique for each VLAN because the appending 48-bit MAC address was unique for each VLAN.

MAC address reduction splits the 16- bit field into 2 fields: a configurable, 4-bit field and a non-configurable, 12-bit field. The non-configurable, 12-bit field carries the VLAN ID or, with MSTP, the MST instance number. The 2 fields are merged to create the unique Bridge Priority for a particular VLAN or, in this case, an MST instance. The appending MAC address remains the same for all instances.

Note MSTP switches must function as if MAC address reduction is enabled.

MAC address reduction is now part of the IEEE 802.1t standard.

Figure 4-5 BPDU Formats



Note When MAC address reduction is enabled, the default root bridge and secondary root bridge priorities are 24576 and 28672 (respectively) instead of 8192 and 16384 (used in Cisco's implementation of 802.1D). The default bridge priority remains unchanged at 32768.

Example 4-5 shows MSTP instance 10, which caused the default root priority of 24576 to increment by 10, making it 24586.

Example 4-5 MSTP Configuration

```
Switch (enable) show spantree mst 10
```

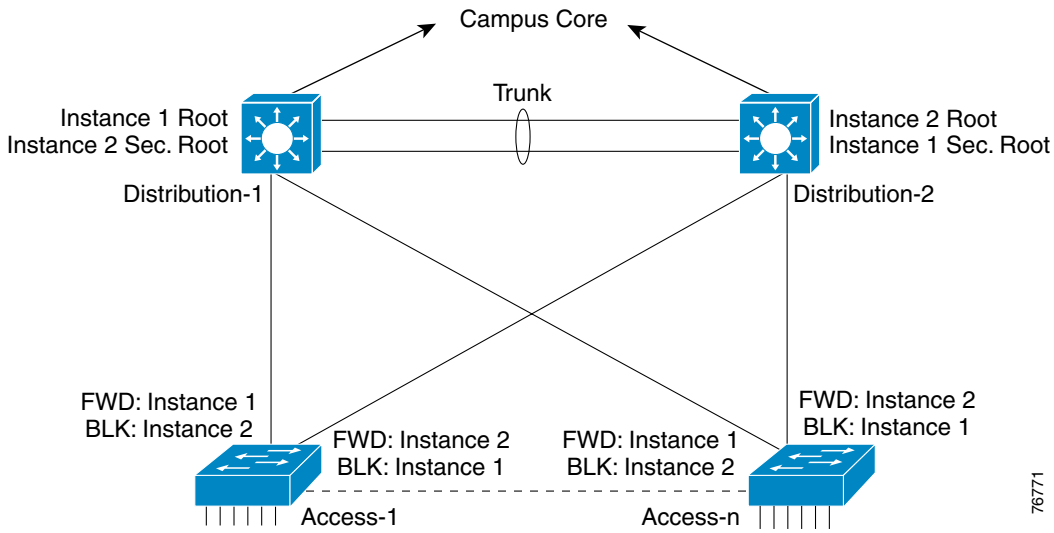
```
Spanning tree mode          MST
Instance                    10
VLANs Mapped:               40
Designated Root             00-d0-04-ae-94-00
Designated Root Priority    24586  (root priority:24576, sys ID ext: 10)
!
!
!
Designated Root Cost        0
Remaining Hops 20
Designated Root Port        1/0
Bridge ID MAC ADDR          00-d0-04-ae-94-00
Bridge ID Priority          24586  (bridge priority: 24576, sys ID ext: 10)
```

Default root bridge priority + 10. MST instance number is 10 in this example.

Configuring MSTP at the Distribution Level

Typically, Distribution 1 or Distribution 2 is set as the spanning root or secondary root, while access switches are left with default spanning tree parameters. Figure 4-6 shows the topology for this example.

Figure 4-6 Configuration Example Topology



Example 4-6 and Example 4-7 show the configuration for IOS at the distribution level. Example 4-8 and Example 4-9 show the configuration for Catalyst OS at the distribution layer.

Example 4-6 MSTP Distribution Level Configuration—IOS

```
Dist-1-CiscoIOS#config t
Dist-1-CiscoIOS(config)#spanning-tree extend system-id
!
Dist-1-CiscoIOS(config)#spanning-tree mode mst
Dist-1-CiscoIOS(config)#spanning-tree mst configuration
Dist-1-CiscoIOS(config-mst)#instance 1 vlan 1-10
Dist-1-CiscoIOS(config-mst)#instance 2 vlan 11-20
Dist-1-CiscoIOS(config-mst)#name mars
Dist-1-CiscoIOS(config-mst)#revision 10
!
Dist-1-CiscoIOS(config)#spanning-tree mst 1 root primary
Dist-1-CiscoIOS(config)#spanning-tree mst 2 root secondary
Dist-1-CiscoIOS(config-mst)#exit
```

Enables MAC Address reduction, also known as extended system-id in IOS. Set spanning tree mode to MST.

Instance to VLAN mapping. Name and revision have to match on all switches that are part of the same MST region.

Make this switch root for instance 1 and secondary root for instance 2.

Example 4-7 Verifying MSTP Distribution Level Configuration—IOS

```
Dist-1-CiscoIOS#show running-config
--------Excerpt ---------
spanning-tree mode mst
spanning-tree extend system-id
!
spanning-tree mst configuration
  name mars
  revision 10
  instance 1 vlan 1-10
  instance 2 vlan 11-20
!
spanning-tree mst 1 priority 24576            Priority value is configured using the "spanning-tree mst
!                                             <instance> primary" command.
spanning-tree mst 2 priority 28672            Priority value configured using the "spanning-tree mst
!                                             <instance> secondary" command.
Dist-1-CiscoIOS#show spanning-tree mst configuration
Name      [mars]
Revision  10
Instance Vlans mapped
------------------------------------------------------------------------
0         21-4094
1         1-10
2         11-20
```

Example 4-8 MSTP Distribution Level Configuration—Catalyst OS

```
Dist-2-CatOS (enable) set spantree mode mst               Set spanning tree mode to MST
Dist-2-CatOS (enable) set spantree macreduction enable    Enable MAC Address reduction, also
!                                                         known as extended system-id in IOS.
Dist-2-CatOS (enable) set spantree mst config name mars revision 10
Dist-2-CatOS (enable) set spantree mst 1 vlan 1-10
Dist-2-CatOS (enable) set spantree mst 2 vlan 11-20
Dist-2-CatOS (enable) set spantree root mst 2
Dist-2-CatOS (enable) set spantree root secondary mst 1   Make this switch root for instance 2 and
!                                                         secondary root for instance 1
Dist-2-CatOS (enable) set spantree mst config commit      'commit' commits changes
```

Example 4-9 Verifying MSTP Distribution Level Configuration—Catalyst OS

```
Dist-2-CatOS (enable) show config
--------Excerpt ---------
#mac address reduction
set spantree macreduction enable
!
#stp mode
set spantree mode mst
!
#spantree
!
#MST (IEEE 802.1s)
set spantree priority 28672 mst 1             Priority value is the result of "spanning-tree mst
!                                             <instance> primary" command.
set spantree priority 24576 mst 2             Priority value is the result of "spanning-tree mst
!                                             <instance> secondary" command.

#MST Configuration
set spantree mst config name mars revision 10
set spantree mst 0 vlan 21-4094               VLANs are part of instance 0 by default
set spantree mst 1 vlan 1-10
```

```
set spantree mst 2 vlan 11-20
set spantree mst config commit

Dist-2-CatOS (enable) show spantree mst config
Current (NVRAM) MST Region Configuration:
Configuration Name: mars
Revision: 10
Instance VLANs
-------- ----------------------------------------------------------------
IST      21-4094
  1      1-10
  2      11-20
  3      -
  4      -
  5      -
  6      -
  7      -
  8      -
  9      -
 10      -
 11      -
 12      -
 13      -
 14      -
 15      -
```

Configuring MSTP at the Access Layer

Example 4-10 and Example 4-11 show the configuration for IOS at the access layer. Example 4-12 and Example 4-13 show the configuration for Catalyst OS at the access layer.

Example 4-10 MSTP Access Level Configuration—IOS

```
Acc-1-CiscoIOS#config t
Acc-1-CiscoIOS(config)#spanning-tree extend system-id
Acc-1-CiscoIOS(config)#spanning-tree mode mst
Acc-1-CiscoIOS(config)#spanning-tree mst configuration
Acc-1-CiscoIOS(config-mst)#instance 1 vlan 1-10
Acc-1-CiscoIOS(config-mst)#instance 2 vlan 11-20
Acc-1-CiscoIOS(config-mst)#name mars
Acc-1-CiscoIOS(config-mst)#revision 10
Acc-1-CiscoIOS(config-mst)#exit
```

Example 4-11 Verifying MSTP Access Level Configuration—IOS

```
Acc-1-CiscoIOS#show running-config
--------Excerpt ----------
spanning-tree mode mst
spanning-tree extend system-id
!
spanning-tree mst configuration
  name mars
  revision 10
  instance 1 vlan 1-10
  instance 2 vlan 11-20
!
Acc-1-CiscoIOS#show spanning-tree mst configuration
Name      [mars]
Revision  10
```

```
Instance  Vlans mapped
--------------------------------------------------------------------------------
0         21-4094
1         1-10
2         11-20
```

Example 4-12 MSTP Access Level Configuration—Catalyst OS

```
Acc-2-CatOS (enable) set spantree mode mst
Acc-2-CatOS (enable) set spantree macreduction enable
Acc-2-CatOS (enable) set spantree mst config name mars revision 10
Acc-2-CatOS (enable) set spantree mst 1 vlan 1-10
Acc-2-CatOS (enable) set spantree mst 2 vlan 11-20
Acc-2-CatOS (enable) set spantree mst config commit
```

Example 4-13 Verifying MSTP Access Level Configuration—Catalyst OS

```
Acc-2-CatOS (enable) show config
--------Excerpt ----------
#mac address reduction
set spantree macreduction enable
!
#stp modeset spantree mode mst
!
#spantree
!
#MST (IEEE 802.1s)

#MST Configuration
set spantree mst config name mars revision 10
set spantree mst 0 vlan 21-4094
set spantree mst 1 vlan 1-10
set spantree mst 2 vlan 11-20
set spantree mst config commit

Acc-2-CatOS (enable) show spantree mst config
Current (NVRAM) MST Region Configuration:
Configuration Name: mars
Revision: 10
Instance VLANs
-------- ------------------------------------------------------------------
IST      21-4094
  1      1-10
  2      11-20
  3      -
  4      -
  5      -
  6      -
  7      -
  8      -
  9      -
 10      -
 11      -
 12      -
 13      -
 14      -
 15      -
```

Interaction Between STPs

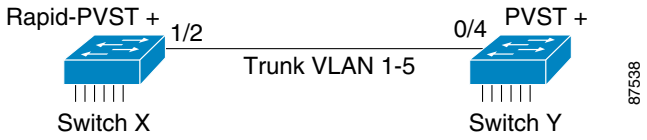
Within a network, switches can be running MSTP, RSTP, PVST+, and Rapid-PVST+. The behavior of the switch may change depending on the spanning-tree protocol running on the other switches with whit it interacts. This section discusses the behavior in the following instances:

- Rapid-PVST+ Interacting with PVST+
- Rapid-PVST+ Interacting with MSTP
- MSTP Interaction (General)
- IST Interacting with STP
- IST Interacting with PVST+
- IST Interacting with 802.1q CST

Rapid-PVST+ Interacting with PVST+

Each RSTP instance will interoperate with the corresponding 802.1D single instance or PVST+. In Figure 4-7, Switch-X is operating in Rapid-PVST+ mode and is connected by a trunk carrying VLANs 1-5 to Switch-Y. Switch-Y is operating in PVST+ mode running an individual instance of 802.1D on VLANs 1 through 5. Switch-X will exchange 802.1D BPDUs on all VLANs of the trunk to seamlessly interact with Switch-Y.

Figure 4-7 Rapid-PVST+ Interacting with PVST+




```
Switch-X#show spanning-tree interface gigabitEthernet 1/2
Vlan             Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- -------------
VLAN0001         Desg FWD 4         128.2    P2p Peer(STP)
VLAN0002         Desg FWD 4         128.2    P2p Peer(STP)
VLAN0003         Desg FWD 4         128.2    P2p Peer(STP)
VLAN0004         Desg FWD 4         128.2    P2p Peer(STP)
VLAN0005         Desg FWD 4         128.2    P2p Peer(STP)

Switch-Y#show spanning-tree interface gigabitEthernet 0/4
Vlan             Port ID                     Designated                 Port ID
Name             Prio.Nbr      Cost Sts      Cost Bridge ID             Prio.Nbr
---------------- -------- --------- --- --------- ----------------------------
VLAN0001         128.4         3004 FWD    200019 32768 00d0.04ae.9400 32.2
VLAN0002         128.4         3004 FWD    200019 32768 00d0.04ae.9400 32.2
VLAN0003         128.4         3004 FWD    200019 32768 00d0.04ae.9400 32.2
VLAN0004         128.4         3004 FWD    200019 32768 00d0.04ae.9400 32.2
VLAN0005         128.4         3004 FWD    200019 32768 00d0.04ae.9400 32.2
```

Rapid-PVST+ Interacting with MSTP

An MSTP switch interacts with a Rapid-PVST+ switch in the same way that an MSTP switch interacts with PVST+ switch. (See IST Interacting with PVST+.) The MSTP switch will send IST BPDUs in 802.1D format on all VLANs to the Rapid-PVST+ switch and IST will consider the port connected to the Rapid-PVST+ switch to be at the boundary of the MST region.

MSTP Interaction (General)

Note This section applies only to networks in which MSTP (802.1s) is deployed.

One of the keys to implementing MSTP is the configuration of the IST instance. IST 0 is an instance that runs on all bridges in an MST region. As discussed in the "IST" section on page 3-7, a very important characteristic of the IST instance is that it provides interaction at the boundary of the MST region with other MST regions. More importantly, the IST is responsible for providing compatibility between the MST regions and other STPs, such as 802.1D, 802.1q (CST), and PVST, connected to the region.To this end, the IST must include timers to interoperate with the other STPs.

Example 4-14 Checking IST Status—IOS

```
Cisco-IOS #show spanning-tree mst 0
###### MST00        vlans mapped:   5-4094
Bridge      address 0002.b940.5b00  priority  4096  (4096 sysid 0)
Root        this switch for CST and IST
Configured  hello time 2, forward delay 15, max age 20, max hops 20
```

Example 4-15 Checking IST Status—Catalyst OS

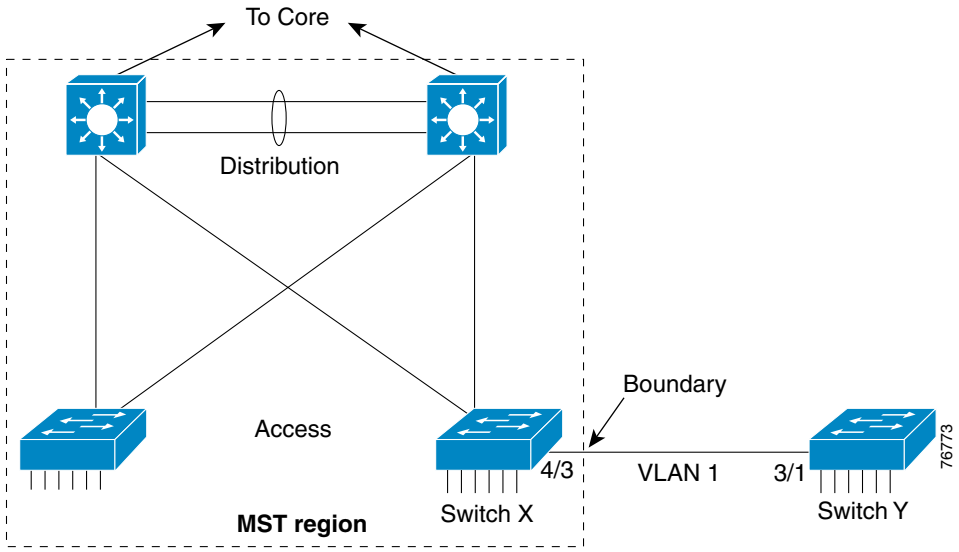
```
CatOS (enable) show spantree mst 0
Spanning tree mode          MST
Instance                    0
VLANs Mapped:               21-4094

Designated Root             00-d0-02-1f-a3-20
Designated Root Priority    4999  (root priority: 4096, sys ID ext: 903)
Designated Root Cost        200019
Designated Root Port        3/1
Root Max Age   20 sec   Hello Time 2  sec   Forward Delay 15 sec

IST Master ID MAC ADDR      00-d0-04-ae-94-00
IST Master ID Priority      32768
IST Master Path Cost        0          Remaining Hops 20
Bridge ID MAC ADDR          00-d0-04-ae-94-00
Bridge ID Priority          32768  (bridge priority: 32768, sys ID ext: 0)
Bridge Max Age 20 sec   Hello Time 2  sec   Forward Delay 15 sec  Max Hops 20
```

IST Interacting with STP

In Figure 4-8, Switch X in the MST region is connected to Switch Y running 802.1D. Ports 4/3 and 3/1 are both in VLAN 1. VLAN 1 is mapped to instance 1 on Switch X. Switch Y is running a single instance of STP. 4/3 is the boundary port of Switch X. IST at the boundary will interact with Switch Y's spanning tree. instance 0 (IST) will interact with Switch Y's VLAN 1 spanning tree because IST alone is

responsible for sending and receiving BPDUs at the boundary of an MST region. Instance 1 is not receiving BPDUs at the boundary. The interaction will be based on 802.1D BPDUs because Switch Y does not understand RSTP. It is also recommended the STP root (Switch Y's root) be configured inside the MST region. In other words, make IST (instance 0) the root for VLAN 1.

Figure 4-8 IST Interacting with Single Instance of 802.1D Spanning Tree



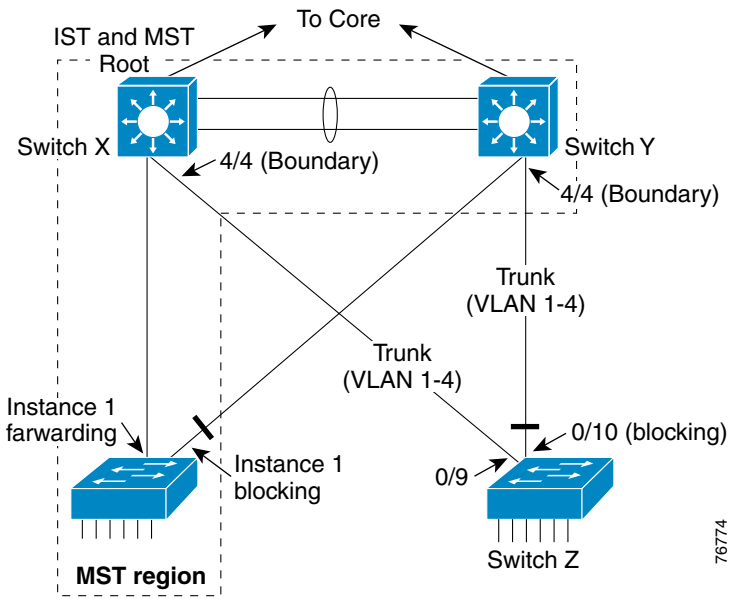
Example 4-16 IST Configuration

```
Switch-X# show spanning-tree interface gig 4/3
Mst Instance      Role Sts Cost      Prio.Nbr Type
----------------- ---- --- --------- -------- --------------------------------
MST00             Desg FWD 20000     128.195  P2p Bound(STP)
MST01             Boun FWD 20000     128.195  P2p Bound(STP)

Switch-Y (enable) show spantree 3/1
Port      Vlan  Port-State     Cost   Priority  Fast-Start  Group-Method
--------- ----  -------------  -----  --------  ----------  ------------
3/1       1     forwarding         4        32   disabled
```

IST Interacting with PVST+

In Figure 4-9, Switches X and Y are in the MST region. VLAN 1-4 is mapped to Instance 1. Port 4/4 is an 802.1q trunk carrying those VLANs at the boundary of the region. Switch Z interface 0/9 and 0/10 are also trunks but the switch is configured for 802.1D. Cisco runs per-VLAN STP (PVSTP+) on trunks. Therefore, Switch Z runs an individual instance of spanning tree for VLANs 1-4. Upon receiving PVST+ BPDUs, the boundary switches will realize that they have a PVST+ speaking neighbor. IST at the boundary of Switch X and Y will replicate (will transmit) IST BPDUs on all VLANs (VLAN 1-4) to be compatible with the neighbors spanning tree instances. It is recommended to simulate the root for VLANs 1-4 inside the MST region. In other words, make IST (instance 0) the root for VLANs 1-4. Doing so will also put the redundant link, 0/10 into blocking state on Switch Z.

Figure 4-9 IST interacting with 802.1D PVST+



Example 4-17 IST Configuration

```
Switch-X#show interface trunk
Port        Mode         Encapsulation  Status        Native vlan
Gi4/4       auto         802.1q          trunking      1

Port        Vlans allowed on trunk
Gi4/4       1-4094

Port        Vlans allowed and active in management domain
Gi4/4       1-4,1002-1005

Port        Vlans in spanning tree forwarding state and not pruned
Gi4/4       1-4,1002-1005

Switch-Y#show interface trunk
Port        Mode         Encapsulation  Status        Native vlan
Gi4/4       auto         802.1q          trunking      1

Port        Vlans allowed on trunk
Gi4/4       1-4094

Port        Vlans allowed and active in management domain
Gi4/4       1-4,1002-1005

Port        Vlans in spanning tree forwarding state and not pruned
Gi4/4       1-4,1002-1005

Switch-X#show spanning-tree interface gig 4/4
Mst Instance      Role Sts Cost      Prio.Nbr Type
----------------- ---- --- --------- -------- --------------------------------
MST00             Desg FWD 20000     128.196  P2p Bound(PVST)
MST01             Boun FWD 20000     128.196  P2p Bound(PVST)
```

```
Switch-Y#show spanning-tree interface gig 4/4
Mst Instance     Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
MST00            Desg FWD 20000     128.196  P2p Bound(PVST)
MST01            Boun FWD 20000     128.196  P2p Bound(PVST)

Switch-Z#show interface trunk
Port      Mode         Encapsulation  Status        Native vlan
Gi0/9     on           802.1q         trunking      1
Gi0/10    on           802.1q         trunking      1

Port      Vlans allowed on trunk
Gi0/9     1-4094
Gi0/10    1-4094

Port      Vlans allowed and active in management domain
Gi0/9     1-4
Gi0/10    1-4

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/9     1-4
Gi0/10    none                                 "none" because Gi0/10 is blocking for VLANs 1-4

Switch-Z#show spanning-tree interface gig 0/9
Vlan             Port ID                    DesignatedPort ID
Name             Prio.Nbr      Cost Sts     Cost Bridge ID Prio.Nbr
---------------- -------- --------- --- --------- -------------------- --------
VLAN0001         128.9             4 FWD     20038 32768 0002.b940.5b00 128.196
VLAN0002         128.9             4 FWD     20038 32768 0002.b940.5b00 128.196
VLAN0003         128.9             4 FWD     20038 32768 0002.b940.5b00 128.196
VLAN0004         128.9             4 FWD     20038 32768 0002.b940.5b00 128.196

Switch-Z#show spanning-tree interface gig 0/10
Vlan             Port ID                    Designated       Port ID
Name             Prio.Nbr      Cost Sts     Cost Bridge ID    Prio.Nbr
---------------- -------- --------- --- --------- --------------------
VLAN0001         128.10            4 BLK     20038 32768 0008.2185.bc00 128.196
VLAN0002         128.10            4 BLK     20038 32768 0008.2185.bc00 128.196
VLAN0003         128.10            4 BLK     20038 32768 0008.2185.bc00 128.196
VLAN0004         128.10            4 BLK     20038 32768 0008.2185.bc00 128.196
```

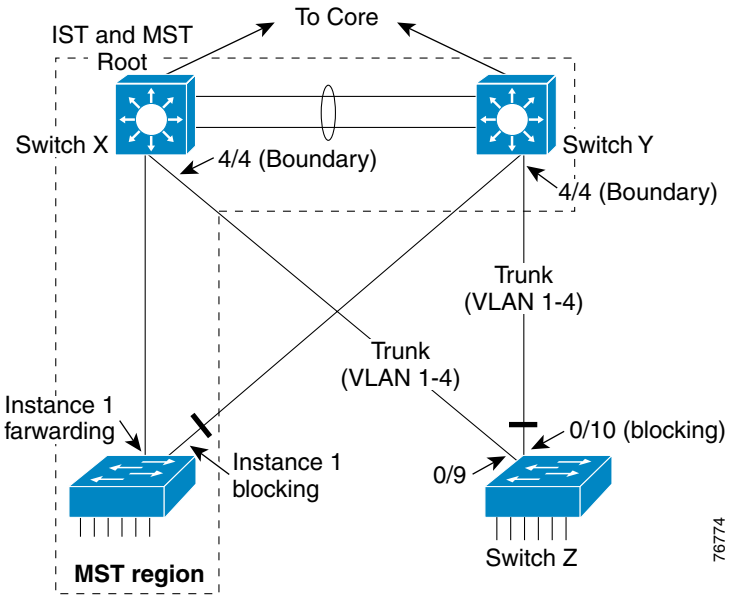
Note The command output above does not illustrate the unmarked access layer switch

Switch-Z can be configured for UplinkFast for fast failover of the root port. For example, Gi0/10 will move to forwarding from blocking, skipping listening and learning if Gi0/9 fails. This can only be possible if the IST is the root for VLANs 1-4.

IST Interacting with 802.1q CST

The 802.1q standard runs one instance of spanning tree for all VLAN (CST). As mentioned before, Cisco's implementation of 802.1q runs individual instances of spanning tree on all VLANs (PVST+). MST regions interact with CST only when it interacts with a third-party switch. IST BPDU interaction with CST is fairly straightforward. IST at the boundary (IST 0) does not need to replicate BPDUs on all VLANs of the trunk because the BPDUs coming from the switch running CST are sent untagged. Hence, IST simply needs to interact with that BPDU.

Figure 4-10 IST interacting with CST



In this configuration:

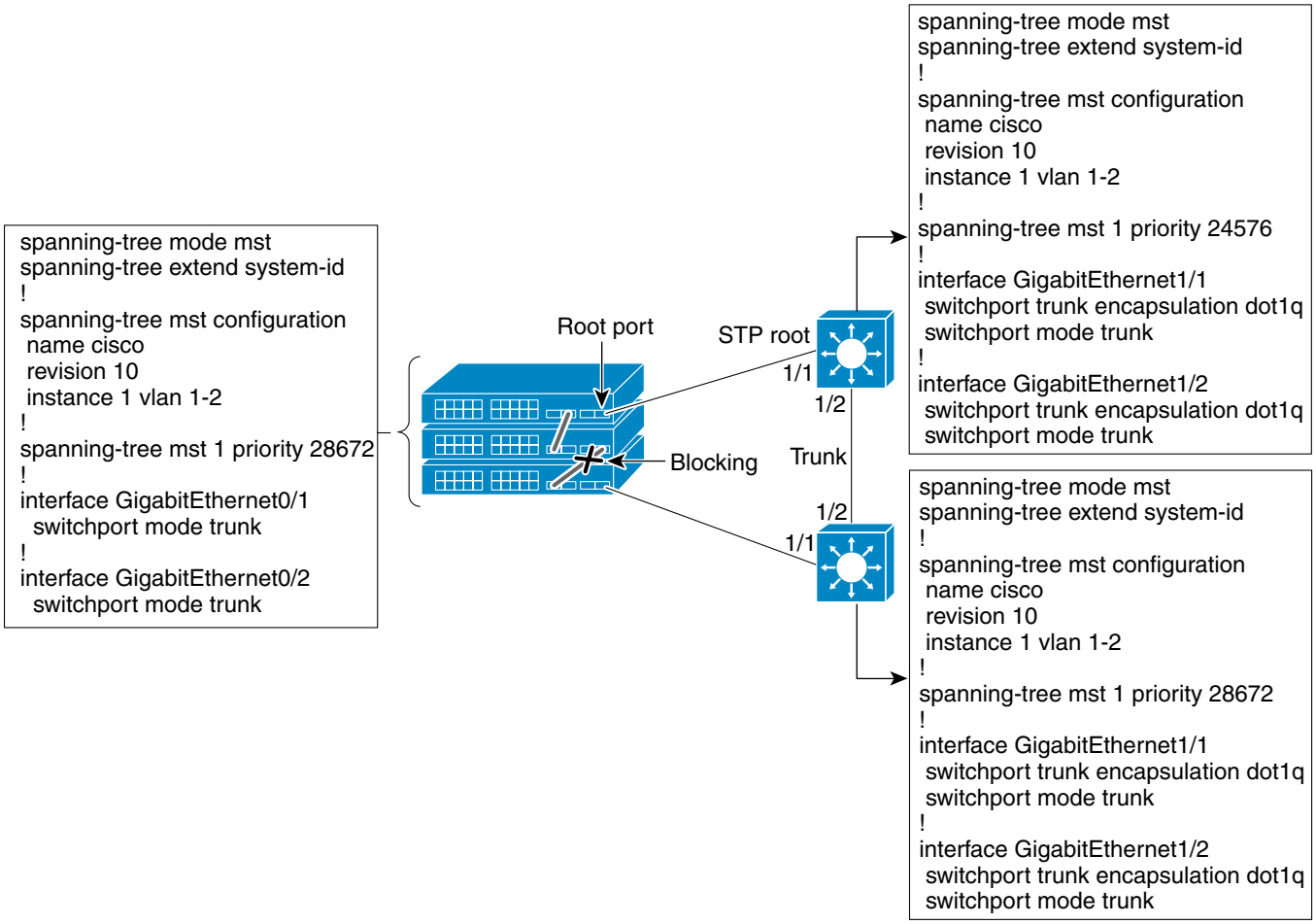
- IST simulates a CST root bridge inside the MST region.
- VLANs are not mapped to IST (instance 0). For more information, see the "Common Misconfigurations" section on page 3-13.

Note If interoperating with an 802.1D network, then it is recommended that the 802.1D network be at the edge of the MST region as in Figure 4-8 and Figure 4-9.

RSTP in a Stack

Rapid transition in a stack works as long as all links in the stack are point-to-point full duplex.

Figure 4-11 RSTP in a Stack



Unlike the Cross-stack UplinkFast (CSUF) feature for stackable switches, RSTP cannot guarantee that the stack ports will not be blocked. CSUF is not supported in 802.1w/s. However, fast transition is still possible.

Upon failure of the root port in Figure 4-11, the blocking link on the middle stack switch will almost immediately transition to forwarding. Unlike CSUF, it is not a requirement to use Gigastack GBICs in half-duplex mode on the middle switch anymore.

Link Type

If both the links in a Gigastack are used, then the interface duplex setting is automatically set to half-duplex. RSTP only accomplishes rapid transition on point-to-point, full-duplex links. A half-duplex link is a shared link from RSTP's perspective and RSTP will fall back to slow transition on this link. The link type determination is based on duplex setting, which can be overwritten but is not recommended on half-duplex stack ports.

```
Cisco-IOS(config-if)#spanning-tree link-type ?
  point-to-point  Consider the interface as point-to-point
  shared          Consider the interface as shared
```

```
CatOS (enable) set spantree mst link-type 3/2 ?
  Auto                      Derive link type from duplex status
  point-to-point            Link type is point-to-point
  shared                    Link type is shared
```

Note Do not override the link type on GigaStack links to achieve fast convergence in RSTP.

Migration Strategy

Keep the following in mind when planning a migration to RSTP or MSTP:

- Properly identify point-to-point and edge ports. Ensure all switch-to-switch links on which a rapid transition is desired are full-duplex. Edge ports are defined through the PortFast feature.
- If using MSTP, carefully decide how many instances will be needed in the switched network, remembering that an instance translates to a logical topology.
- If using MSTP, decide what VLANs to map onto those instances and carefully select a root and a backup root for each instance.
- Choose a configuration name and a revision number that will be common to all switches in the network.
- Cisco recommends placing as many switches as possible into a single region; there is no advantage in segmenting a network in separate regions.
- Always try to keep the root of the CST/IST inside the region.
- Avoid mapping any VLANs onto instance 0.
- Converting spanning tree to RSTP and MST from 802.1D on an existing switched network can cause traffic outage till the configuration is complete on all switches. Converting switches independently will disrupt network continuity if the switch is in the middle of an MST region. Therefore, MST configuration in a production network should be carried out during the maintenance window.
- Start by migrating the core first by changing the STP type to MSTP, and work your way down to the access switches. MSTP can interact with legacy bridges running PVST+ on a per-port basis, so it is not a problem to mix both types of bridges if interactions are clearly understood.
- If an MST region is present already, then adding new switches around it is fairly non-disruptive.
- MST interoperates well with legacy spanning tree.
  - Follow the examples in the "MSTP Interaction (General)" section if the plan is to first interoperate with legacy spanning tree then change the spanning tree mode to MST.
  - When attaching an 802.1D switch to an MST region, ensure that the root for all VLANs configured on the switch is inside the MST region. In other words, ensure that IST is the root for all VLANs in the 802.1D region. Once the interaction is established, the new switch can be converted to MST at a later time. If there is a need to interoperate with legacy 802.1D network, then put the 802.1D network at the edge of the MST region.
  - When interacting with a PVST+ bridge through a trunk, ensure the MSTP bridge is the root for all VLANs allowed on that trunk.

Spanning Tree Logical Ports

The sum of all logical ports equals the number of trunks on the switch times number of active VLANs on the trunks, plus the number of non-trunking access ports on the switch. Table 4-1 lists the maximum supported logical ports across platforms and minimum release required to run RSTP and MSTP.

Table 4-1 Logical Ports

		MST: Maximum logical ports	
Platform	**Release**	**Per switching module**	**Total**
Catalyst 6500 SUP 2	Cisco IOS 12.1(11b)EX1	6000	24000
	Catalyst OS 7.1		127000
Catalyst 6500 SUP 1	Cisco IOS 12.1(11b)EX1	3000	12000
	Catalyst OS 7.1		40000
Catalyst 4006 SUP 3/4	Cisco IOS 12.1(12c)EW	3000	9000
Catalyst 400x SUP 1/2	Catalyst OS 7.1		9000
Catalyst 3550	12.1(9)EA1	N/A	20000
Catalyst 2950	12.1(9)EA1	N/A	20000

Note Rapid-Per VLAN STP is supported in IOS 12.1(13E) for Catalyst 6000 and Catalyst OS 7.5.1

To verify the number of logical ports supported, issue one of the following commands:

```
CatOS (enable) show spantree summ novlan

Cisco-IOS #show spanning-tree summary totals
```

Spanning Tree Extensions

MST and Rapid-PVST+ support the following Cisco 802.1D extensions:

- Spanning-Tree PortFast, BPDU Guard, and BPDU Filtering
- Spanning-Tree Loop Guard

Spanning-Tree PortFast, BPDU Guard, and BPDU Filtering

The spanning-tree PortFast, BPDU Guard, and BPDU filtering features are all inter-related.

PortFast

Because spanning-tree categorizes ports into edge and non-edge, which is based on the duplex information as well as the assignment of PortFast to a port, it is important to configure the PortFast feature on all eligible ports. PortFast transitions the port directly into forwarding after linkup rather than going through the spanning tree transition states, which delay link bring up by forward delay time (15 seconds, by default) at each transition. This makes the network more stable because it keeps a port in

forwarding state during topology changes. Failure to configure PortFast has drastic effects on the convergence time: a non-edge port connected to a device that does not speak spanning-tree cannot perform the handshake that shortens convergence time. Consequently, a non-edge port connected to a server or a service appliance goes through the blocking, learning, and forwarding steps slowing down the convergence time by 30 seconds. This is still acceptable if it happens on a single server port (meaning this single server is going to be unavailable for 30 seconds). However, this has major effects if all of the servers in the server farm have to go through this process and/or if the service modules are affected by this delay (all the traffic has to traverse these modules).

Example 4-18 Configuring PortFast—IOS

```
Cisco-IOS#configure terminal
Cisco-IOS(config)#interface FastEthernet 0/4
Cisco-IOS(config-if)# spanning-tree portfast
```

Example 4-19 Configuring PortFast—Catalyst OS

```
CatOS> (enable) set spantree portfast 4/1 enable
```

In RSTP, if an edge port receives a BPDU, it will lose its edge status and revert to being a normal spanning tree port.

Example 4-20 Verifying Port Status—Before Receiving a BPDU

```
Swtich (enable) show spantree 3/2
Port                     Vlan Port-State    Cost      Prio Portfast Channel_id
------------------------ ---- ------------- --------- ---- -------- ----------
 3/2                     1    forwarding       200000   32 enabled  0
Swtich (enable) show spantree mst 3/2
--------Excerpt ----------
Edge Port:          Yes, (Configured) Enable
Link Type:          P2P, (Configured) Auto
Port Guard:     Default
Boundary:       No
Swtich (enable) show spantree mst 1
--------Excerpt ----------
Port                     State         Role Cost     Prio Type
------------------------ ------------- ---- -------- ---- --------------------
 3/2                     forwarding    DESG   200000   32 P2P, Edge
```

Example 4-21 Verifying Port Status—After Receiving a BPDU

```
Swtich (enable) show spantree mst 3/2
--------Excerpt ----------
Edge Port:           No, (Configured) Enable
Link Type:          P2P, (Configured) Auto
Port Guard:     Default
Boundary:       Yes (STP)
Swtich (enable) show spantree mst 1
Port                     State         Role Cost     Prio Type
------------------------ ------------- ---- -------- ---- --------------------
3/2                     forwarding    BDRY   200000   32 P2P, Boundary(STP)
```

BPDU Guard

In a valid configuration, PortFast enabled ports are connected to edge devices and do not receive BPDUs. Receiving a BPDU on a PortFast enabled port indicates connection of an unauthorized device to the edge port. BPDU Guard protects the network by disabling the port if a misconfiguration caused an access port to receive a BPDU. When enabled globally, BPDU Guard applies to all PortFast enabled interfaces.

Example 4-22 Configuring BPDU Guard—IOS

```
Cisco-IOS#spanning-tree portfast bpduguard
```

Example 4-23 Configuring BPDU Guard—Catalyst OS

```
CatOS> (enable) set spantree portfast bpdu-guard enable
```

BPDU Guard can also be applied at interface level. Interface configuration overrides global configuration.

Example 4-24 Configuring BPDU Guard at the Interface—IOS

```
Cisco-IOS#configure terminal
Cisco-IOS(config)#interface gigabitEthernet 0/4
Cisco-IOS(config-if)#spanning-tree bpduguard enable
```

Example 4-25 Configuring BPDU Guard at the Interface—Catalyst OS

```
CatOS (enable) set spantree portfast bpdu-guard 4/1 enable
```

BPDU Filtering

BPDU Filtering stops a port from transmitting BPDUs on a port connected to an end system. When enabled globally, BPDU Filtering applies to all Portfast enabled interfaces.

Example 4-26 Configuring BPDU Filtering—IOS

```
Cisco-IOS(config)#spanning-tree portfast bpdufilter default
```

Example 4-27 Configuring BPDU Filtering—Catalyst OS

```
CatOS (enable) set spantree portfast bpdu-filter enable
```

BPDU Filtering can also be applied at interface level. Interface configuration overrides global configuration.

Example 4-28 Configuring BPDU Filtering at the Interface—IOS

```
Cisco-IOS#configure terminal
Cisco-IOS(config)#interface gigabitEthernet 0/4
Cisco-IOS(config-if)#spanning-tree bpdufilter enable
```

Example 4-29 Configuring BPDU Filtering at the Interface—Catalyst OS

```
CatOS (enable) set spantree portfast bpdu-filter 4/1 enable
```

Spanning-Tree Loop Guard

Unidirectional link failures can cause a root port or alternate port to stop receiving BPDUs. The absence of BPDUs on a switch port can lead to spanning tree loops. Loop Guard prevents alternate or root port to become designated as root in the absence of BPDUs by determining whether a root port or an alternate root port is receiving BPDUs. If the port is not receiving BPDUs, the loop guard feature will put the port into an inconsistent state. The loop guard feature is effective on both alternate (blocking) and root port.

Loop Guard can also be safely enabled globally on all ports using a global configuration command. The global configuration might be simpler and more efficient as there is no need to re-configure the specific uplink if there is a cabling change.

Example 4-30 Configuring Spanning-Tree Loop Guard—IOS

```
Cisco-IOS#configure terminal
Cisco-IOS(config)#interface gigabitEthernet 0/4
Cisco-IOS(config-if)#spanning-tree guard loop
Cisco-IOS(config)#interface gigabitEthernet 0/5
Cisco-IOS(config-if)#spanning-tree guard loop        One of the uplinks will be in blocking state for
!                                                      the MST instance if the root bridge is on the
!                                                      distribution switch
Cisco-IOS(config)#spanning-tree loopguard default    Enables loop guard globally
```

Example 4-31 Configuring Spanning-Tree Loop Guard—Catalyst OS

```
CatOS (enable) set spantree guard loop 4/4-5
CatOS (enable) set spantree global-default portfast enable      Enables loop guard globally
```