Multiple Spanning Tree Protocol

This chapter introduces you to Multiple Spanning Tree Protocol (MSTP) which is one of the variants of Spanning Tree Protocol (STP) and describes how you can configure the MSTP feature.

Multiple Spanning Tree Protocol

The Multiple Spanning Tree Protocol (MSTP) is a Spanning Tree Protocols (STPs) variant that allows you to create multiple and independent spanning trees over the same physical network. You can configure the parameters for each spanning tree separately. You can select different network devices as the root bridge or different paths to form the loop-free topology. Therefore, you can block a given physical interface for some of the spanning trees and unblock for others.

After setting up multiple spanning tree instances, you can partition the set of VLANs in use. For example, you can assign VLANs 1–100 to spanning tree instance 1, VLANs 101–200 to spanning tree instance 2, VLANs 201–300 to spanning tree instance 3, and so on. Since each spanning tree has a different active topology with different active links, this has the effect of dividing the data traffic among the available redundant links based on the VLAN—a form of load balancing.

MSTP Supported Features

The Cisco 8000 Series Routers support MSTP, as defined in IEEE 802.1Q-2005, on physical Ethernet interfaces and Ethernet Bundle interfaces.This includes the Port Fast and bridge protocol data unit (BPDU) Guard features to break L2 loop. The routers can operate in either standard 802.1Q mode, or in Provide Edge (802.1ad) mode. In provider edge mode, a different MAC address is used for bridge protocol data units (BPDUs), and any BPDUs received with the 802.1Q MAC address are forwarded transparently.

When you have not configured the **allow-bpdu-guard** command on MST default instance, and if one of the bridge ports receives legacy BPDU, the port enters **error-disable** state.

BPDU Guard

The BPDU Guard feature allows you to protect against misconfiguration of edge ports. It is an enhancement to the MSTP port fast feature. When you configure port fast on an interface, MSTP considers that interface to be an edge port and removes it from consideration when calculating the spanning tree. When you configure BPDU Guard, MSTP additionally shuts down the interface using error-disable when an MSTP BPDU is received.

Restrictions

These restrictions apply when using MSTP:

- You can configure MSTP only on the main (L3 or L2 interface) interface.
- The subinterfaces are mapped to MSTI instances by the outermost VLAN tag ID, even if the subinterface encapsulation is a QinQ or a single VLAN tag.
- There's no intersection with split-horizon group alignment and MSTI grouping using VLAN ID. Each grouping runs independently.
- All subinterfaces in a bridge domain must use the same MSTI when MSTP is running on the corresponding main interfaces.
- When MSTP runs on a main interface, untagged subinterface shouldn't be created.

Configure MSTP

By default, STP is disabled on all interfaces. You must enable MSTP on each physical or Ethernet Bundle interface. When you configure MSTP on an interface, all the subinterfaces of that interface are automatically MSTP-enabled.

Perform these tasks to configure MSTP:

- Configure VLAN interfaces
- Configure L2VPN bridge-domains
- Configure MSTP parameters

Configuration Example

```
/* Configure VLAN interfaces */
Router# configure
Router(config)# interface HundredGigE0/0/0/2.1001 l2transport
Router(config-subif)# encapsulation dot1q 1001
Router(config)# interface HundredGigE0/0/0/3.1001 l2transport
Router(config-subif)# encapsulation dot1q 1001
Router(config)# interface HundredGigE0/0/0/14.1001 l2transport
Router(config-subif)# encapsulation dot1q 1001
Router(config)# interface HundredGigE0/0/0/2.1021 l2transport
Router(config-subif)# encapsulation dot1q 1021
Router(config)# interface HundredGigE0/0/0/3.1021 l2transport
```

```
Router(config-subif)# encapsulation dot1q 1021
Router(config)# interface HundredGigE0/0/0/14.1021 l2transport
Router(config-subif)# encapsulation dot1q 1021
Router(config-subif)# commit

/* Configure L2VPN bridge-domains */
Router# configure
Router(config)# l2vpn bridge group mstp
Router(config-l2vpn-bg)# bridge-domain mstp1001
Router(config-l2vpn-bg-bd)# int HundredGigE 0/0/0/2.1001
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# int HundredGigE 0/0/0/3.1001
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# int HundredGigE 0/0/0/14.1001
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# exit
Router(config-l2vpn-bg)# exit
Router(config-l2vpn-bg)# bridge-domain mstp1021
Router(config-l2vpn-bg-bd)# int HundredGigE 0/0/0/2.1021
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# int HundredGigE 0/0/0/3.1021
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# int HundredGigE 0/0/0/14.1021
Router(config-l2vpn-bg-bd-ac)# commit

/* Configure MSTP Parameters */
Router#configure
Router(config)#spanning-tree mst a
Router(config-mstp)#interface HundredGigE0/0/0/2
Router(config-mstp-if)#portfast bpduguard
Router(config-mstp-if)#commit
```

Running Configuration

This section show MSTP running configuration.

```
!
Configure
/* Configure VLAN interfaces */
interface HundredGigE0/0/0/2.1001 l2transport
 encapsulation dot1q 1001
!
interface HundredGigE0/0/0/3.1001 l2transport
 encapsulation dot1q 1001
!
interface HundredGigE0/0/0/14.1001 l2transport
 encapsulation dot1q 1001

interface HundredGigE0/0/0/2.1021 l2transport
 encapsulation dot1q 1021
!
interface HundredGigE0/0/0/3.1021
 l2transport
 encapsulation dot1q 1021
!
interface HundredGigE0/0/0/14.1021 l2transport
 encapsulation dot1q 1021
!
/* Configure L2VPN Bridge-domains */
l2vpn
 bridge group mstp
  bridge-domain mstp1001
```

```
   interface HundredGigE0/0/0/2.1001
   !
   interface HundredGigE0/0/0/3.1001
   !
   interface HundredGigE0/0/0/14.1001
   !
bridge-domain mstp1021
   interface HundredGigE0/0/0/2.1021
   !
   interface HundredGigE0/0/0/3.1021
   !
   interface HundredGigE0/0/0/14.1021
!
/* Configure MSTP Parameters */
spanning-tree mst a
 interface HundredGigE0/0/0/2
   portfast bpduguard
!
!
```

Information About Multiple Spanning Tree Protocol

To configure Ethernet services access lists, you must understand these concepts:

Spanning Tree Protocol Overview

Ethernet is no longer just a link-layer technology used to interconnect network vehicles and hosts. Its low cost and wide spectrum of bandwidth capabilities coupled with a simple *plug and play* provisioning philosophy have transformed Ethernet into a legitimate technique for building networks, particularly in the access and aggregation regions of service provider networks.

Ethernet networks lacking a TTL field in the Layer 2 (L2) header and, encouraging or requiring multicast traffic network-wide, are susceptible to broadcast storms if loops are introduced. However, loops are a desirable property as they provide redundant paths. Spanning tree protocols (STP) are used to provide a loop free topology within Ethernet networks, allowing redundancy within the network to deal with link failures.

There are many variants of STP; however, they work on the same basic principle. Within a network that may contain loops, a sufficient number of interfaces are disabled by STP so as to ensure that there is a loop-free spanning tree, that is, there is exactly one path between any two devices in the network. If there is a fault in the network that affects one of the active links, the protocol recalculates the spanning tree so as to ensure that all devices continue to be reachable. STP is transparent to end stations which cannot detect whether they are connected to a single LAN segment or to a switched LAN containing multiple segments and using STP to ensure there are no loops.

STP Protocol Operation

All variants of STP operate in a similar fashion: STP frames (known as bridge protocol data units (BPDUs)) are exchanged at regular intervals over Layer 2 LAN segments, between network devices participating in STP. Such network devices do not forward these frames, but use the information to construct a loop free spanning tree.

The spanning tree is constructed by first selecting a device which is the *root* of the spanning tree (known as the root bridge), and then by determining a loop free path from the *root bridge* to every other device in the network. Redundant paths are disabled by setting the appropriate ports into a blocked state, where STP frames can still be exchanged but data traffic is never forwarded. If a network segment fails and a redundant path

exists, the STP protocol recalculates the spanning tree topology and activates the redundant path, by unblocking the appropriate ports.

The selection of the root bridge within a STP network is determined by the lowest Bridge ID which is a combination of configured bridge priority and embedded mac address of each device. The device with the lowest priority, or with equal lowest priority but the lowest MAC address is selected as the root bridge.

Root port: is selected based on lowest root path cost to root bridge. If there is a tie with respect to the root path cost, port on local switch which receives BPDU with lowest sender bridge ID is selected as root port.

Designated port: Least cost port on local switch towards root bridge is selected as designated port. If there is a tie, lowest number port on local switch is selected as designated port.

The selection of the active path among a set of redundant paths is determined primarily by the port path cost. The port path cost represents the cost of transiting between that port and the root bridge - the further the port is from the root bridge, the higher the cost. The cost is incremented for each link in the path, by an amount that is (by default) dependent on the media speed. Where two paths from a given LAN segment have an equal cost, the selection is further determined by the lowest bridge ID of the attached devices, and in the case of two attachments to the same device, by the configured port priority and port ID of the neighboring attached ports.

Once the active paths have been selected, any ports that do not form part of the active topology are moved to the blocking state.

Variants of STP

The following are the supported variants of the Spanning Tree Protocol:

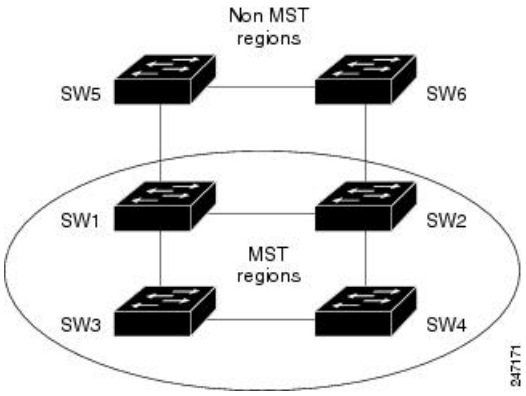
- Legacy STP (STP)—The original STP protocol was defined in IEEE 802.1D-1998. This creates a single spanning tree which is used for all VLANs and most of the convergence is timer-based.
- Multiple STP (MSTP)—A further enhancement was defined in IEEE 802.1Q-2005. This allows multiple spanning tree instances to be created over the same physical topology. By assigning different VLANs to the different spanning tree instances, data traffic can be load-balanced over different physical links. The number of different spanning tree instances that can be created is restricted to a much smaller number than the number of possible VLANs; however, multiple VLANs can be assigned to the same spanning tree instance. The BPDUs used to exchange MSTP information are always sent untagged; the VLAN and spanning tree instance data is encoded inside the BPDU.

MSTP Regions

Along with supporting multiple spanning trees, MSTP also introduces the concept of regions. A region is a group of devices under the same administrative control and have similar configuration. In particular, the configuration for the region name, revision, and the mapping of VLANs to spanning tree instances must be identical on all the network devices in the region. A digest of this information is included in the BPDUs sent by each device, so as to allow other devices to verify whether they are in the same region.

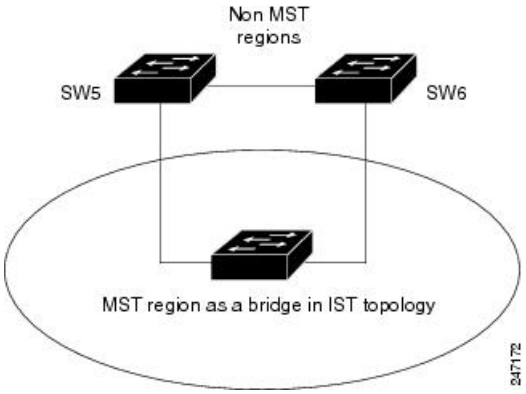
The following figure shows the operation of MST regions when bridges running MSTP are connected to bridges running legacy STP or RSTP. In this example, switches SW1, SW2, SW3, SW4 support MSTP, while switches SW5 and SW6 do not.

Figure 1: MST Interaction with Non-MST Regions



To handle this situation, an Internal Spanning Tree (IST) is used. This is always spanning tree instance 0 (zero). When communicating with non-MSTP-aware devices, the entire MSTP region is represented as a single switch. The logical IST topology in this case is shown in the following figure.

Figure 2: Logical Topology in MST Region Interacting with Non-MST Bridges



The same mechanism is used when communicating with MSTP devices in a different region. For example, SW5 in the above figure could represent a number of MSTP devices, all in a different region compared to SW1, SW2, SW3 and SW4.

MSTP Port Fast

MSTP includes a *Port Fast* feature for handling ports at the edge of the switched Ethernet network. For devices that only have one link to the switched network (typically host devices), there is no need to run MSTP, as there is only one available path. Furthermore, it is undesirable to trigger topology changes (and resultant MAC flushes) when the single link fails or is restored, as there is no alternative path.

By default, MSTP monitors ports where no BPDUs are received, and after a timeout, places them into *edge mode* whereby they do not participate in MSTP. However, this process can be speeded up (and convergence of the whole network thereby improved) by explicitly configuring edge ports as port fast.

Note

- You must disable and re-enable the port for Port Fast configuration to take effect. Use **shutdown** and **no shutdown** command (in interface configuration mode) to disable and re-enable the port.
- Port Fast is implemented as a Cisco-proprietary extension in Cisco implementations of legacy STP. However, it is encompassed in the standard for MSTP, where it is known as Edge Port.

MSTP Root Guard

In networks with shared administrative control, it may be desirable for the network administrator to enforce aspects of the network topology and in particular, the location of the root bridge. By default, any device can become the root bridge for a spanning tree, if it has a lower priority or bridge ID. However, a more optimal forwarding topology can be achieved by placing the root bridge at a specific location in the centre of the network.

Note The administrator can set the root bridge priority to 0 in an effort to secure the root bridge position; however, this is no guarantee against another bridge which also has a priority of 0 and has a lower bridge ID.

The root guard feature provides a mechanism that allows the administrator to enforce the location of the root bridge. When root guard is configured on an interface, it prevents that interface from becoming a root port (that is, a port via which the root can be reached). If superior information is received via BPDUs on the interface that would normally cause it to become a root port, it instead becomes a backup or alternate port. In this case, it is placed in the blocking state and no data traffic is forwarded.

The root bridge itself has no root ports. Thus, by configuring root guard on every interface on a device, the administrator forces the device to become the root, and interfaces receiving conflicting information are blocked.

Note Root Guard is implemented as a Cisco-proprietary extension in Cisco implementations of legacy STP. However, it is encompassed in the standard for MSTP, where it is known as Restricted Role.

MSTP Topology Change Guard

In certain situations, it may be desirable to prevent topology changes originating at or received at a given port from being propagated to the rest of the network. This may be the case, for example, when the network is not under a single administrative control and it is desirable to prevent devices external to the core of the network from causing MAC address flushing in the core. This behavior can be enabled by configuring Topology Change Guard on the port.

Note Topology Change Guard is known as *Restricted TCN* in the MSTP standard.