CCIE Enterprise Infrastructure Playbook - Switched Campus

Switch Administration

Managing MAC address table

Reference: Managing the MAC Address Table

The MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address: a source MAC address that the switch learns and then ages when it is not in use.
- Static address: a manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port number associated with the address and the type (static or dynamic).

By default, MAC address learning is enabled on all interfaces and VLANs on the router. You can control MAC address learning on an interface or VLAN to manage the available MAC address table space by controlling which interfaces or VLANs can learn MAC addresses. Before you disable MAC address learning, be sure that you are familiar with the network topology and the router system configuration.

Disabling MAC address learning on an interface or VLAN could cause flooding in the network.

You can disable MAC address learning on a single VLAN ID from 1 to 4094.

Example:

```
no mac address-table learning vlan 223
```

You can also disable MAC address learning on a range of VLAN IDs, separated by hyphen or comma.

Example:

```
no mac address-table learning vlan 1-10, 15
```

It is recommended that you disable MAC address learning only in VLANs with two ports. If you disable MAC address learning on a VLAN with more than two ports, every packet entering the switch is flooded in that VLAN domain.

You cannot disable MAC address learning on a VLAN that is used internally by the router. If the VLAN ID that you enter is an internal VLAN, the switch generates an error message and rejects the command. To view internal VLANs in use, enter the `show vlan internal usage` privileged EXEC command.

If you disable MAC address learning on a VLAN that includes a secure port, MAC address learning is not disabled on that port.

[**no**] **mac-address-table learning** {**vlan** *vlan-id* [,*vlan-id* | -*vlan-id*] | **interface** *interface slot/port*}

Disable/Enable MAC address learning on an interface or on a specified VLAN or VLANs.

You can specify a single VLAN ID or a range of VLAN IDs separated by a hyphen or comma. Valid VLAN IDs 1 to 4094. It cannot be an internal VLAN.

Errdisable recovery

Reference: Errdisable Port State Recovery on the Cisco IOS Platforms

If the configuration shows a port to be enabled, but software on the switch detects an error situation on the port, the software shuts down that port. In other words, the port is automatically disabled by the switch operating system software because of an error condition that is encountered on the port.

When a port is error disabled, it is effectively shut down and no traffic is sent or received on that port. When you issue the show interfaces command, the port status shows err-disabled.

```
cat6knative#show interfaces gigabitethernet 4/1 status

Port     Name         Status        Vlan        Duplex  Speed Type

Gi4/1                 err-disabled 100            full    1000 1000BaseSX
```

Or, if the interface has been disabled because of an error condition, you can see messages that are similar to these in both the console and the syslog.

This example message displays when a host port receives the bridge protocol data unit (BPDU). The actual message depends on the reason for the error condition.

```
%SPANTREE-SP-2-BLOCK_BPDUGUARD:
   Received BPDU on port GigabitEthernet4/1 with BPDU Guard enabled.
Disabling port.

%PM-SP-4-ERR_DISABLE:
   bpduguard error detected on Gi4/1, putting Gi4/1 in err-disable
state
```

There are various reasons for the interface to go into errdisable. The reason can be:

- Duplex mismatch
- Port channel misconfiguration
- BPDU guard violation
- UniDirectional Link Detection (UDLD) condition
- Late-collision detection
- Link-flap detection
- Security violation
- Port Aggregation Protocol (PAgP) flap
- Layer 2 Tunneling Protocol (L2TP) guard

- DHCP snooping rate-limit
- Incorrect GBIC / Small Form-Factor Pluggable (SFP) module or cable
- Address Resolution Protocol (ARP) inspection
- Inline power

Note: Error-disable detection is enabled for all of these reasons by default. In order to disable error-disable detection, use the `no errdisable detect cause` command. The `show errdisable detect` command displays the error-disable detection status.

If you have enabled errdisable recovery, you can determine the reason for the errdisable status if you issue the show errdisable recovery command.

Here is an example:

```
cat6knative#show errdisable recovery
ErrDisable Reason    Timer Status
-----------------    --------------
udld                 Enabled
bpduguard            Enabled
security-violatio    Enabled
channel-misconfig    Enabled
pagp-flap            Enabled
dtp-flap             Enabled
link-flap            Enabled
l2ptguard            Enabled
psecure-violation    Enabled
gbic-invalid         Enabled
dhcp-rate-limit      Enabled
mac-limit            Enabled
unicast-flood        Enabled
arp-inspection       Enabled

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

Interface      Errdisable reason      Time left(sec)
---------    ---------------------    --------------
  Fa2/4                bpduguard          273
```

In order to recover a port from the errdisable state, first identify and correct the root problem, and then re-enable the port. If you re-enable the port before you fix the root problem, the ports just become error disabled again.

After you discover why the ports were disabled, fix the root problem. The fix depends on the triggering problem.

After you fix the root problem, the ports are still disabled if you have not configured errdisable recovery on the switch. In this case, you must re-enable the ports manually. Issue the `shutdown` command and then the `no shutdown` interface mode command on the associated interface in order to manually reenable the ports.

The errdisable recovery command allows you to choose the type of errors that automatically re-enable the ports after a specified amount of time.

Note: The default timeout interval is 300 seconds and, by default, the timeout feature is disabled.

In order to turn on automatic errdisable recovery and choose the errdisable conditions, issue the `errdisable recovery cause` command.

```
cat6knative(config)#errdisable recovery cause bpduguard
```

If any one of the errdisable recovery conditions is enabled, the ports with this condition are reenabled after 300 seconds. You can also change this default value.

This example changes the errdisable recovery interval from 300 to 400 seconds.

```
cat6knative(config)#errdisable recovery interval 400
```

L2 MTU

Reference: [Configuring System MTU]

The default maximum transmission unit (MTU) size for frames received and transmitted on all interfaces is 1500 bytes. You can increase the MTU size for all interfaces operating at 10 or 100 Mb/s by using the `system mtu` global configuration command. You can increase the MTU size to support jumbo frames on all Gigabit Ethernet interfaces by using the `system mtu jumbo` global configuration command.

Gigabit Ethernet ports are not affected by the `system mtu` command; 10/100 ports are not affected by the `system mtu jumbo` command. If you do not configure the `system mtu jumbo` command, the setting of the `system mtu` command applies to all Gigabit Ethernet interfaces.

Example:

```
Switch(config)# system mtu 2500
```

system mtu *bytes*
The range is 1500 to 1998 bytes; the default is 1500 bytes.

Example:

```
Switch(config)# system mtu jumbo 7500
```

system mtu jumbo *bytes*
The range is 1500 to 9198 bytes; the default is 1500 bytes.

Layer 2 Protocols

CDP

Reference: Cisco Discovery Protocol Configuration Guide, Cisco IOS Release 15M&T

Cisco Discovery Protocol is a Layer 2, media-independent, and network-independent protocol that networking applications use to learn about nearby, directly connected devices. Cisco Discovery Protocol is enabled by default. Each device configured for Cisco Discovery Protocol advertises at least one address at which the device can receive messages and sends periodic advertisements (messages) to the well-known multicast address 01:00:0C:CC:CC:CC. Devices discover each other by listening at that address. They also listen to messages to learn when interfaces on other devices are up or go down.

Advertisements supported and configured in Cisco software are sent, by default, every 60 seconds. Cisco devices never forward Cisco Discovery Protocol packets. Cisco devices that support Cisco Discovery Protocol store the information received in a table. Information in this table is refreshed every time an advertisement is received, and information about a device is discarded after three advertisements from that device are missed.

The information contained in Cisco Discovery Protocol advertisements varies based on the type of device and the installed version of the operating system. Some of the information that Cisco Discovery Protocol can learn includes:

- Cisco IOS version running on Cisco devices
- Hardware platform of devices
- IP addresses of interfaces on devices
- Locally connected devices advertising Cisco Discovery Protocol
- Interfaces active on Cisco devices, including encapsulation type
- Hostname
- Duplex setting
- VLAN Trunking Protocol (VTP) domain
- Native VLAN

Cisco Discovery Protocol provides the following benefits:

- Allows systems using different network layer protocols to learn about one another.
- Facilitates management of Cisco devices by discovering them and discovering how they are configured.
- Assists with troubleshooting Type-Length-Value Fields (TLV) fields.
- Works with SNMP by learning SNMP agent addresses and sending SNMP queries.

Cisco Discovery Protocol Version 2 provides more intelligent, device-tracking features than those available in Version 1. The broadcasting of Cisco Discovery Protocol Version 2 advertisements is enabled by default on Cisco devices.

Example: Disabling and Enabling Cisco Discovery Protocol on a Cisco Device

```
Device(config)# [no] cdp run
```

[**no**] **cdp run**
Disables or enables Cisco Discovery Protocol on a supported device.

If the encapsulation of an interface is changed, Cisco Discovery Protocol is reenabled on that interface even if Cisco Discovery Protocol was previously disabled. For example, when interface encapsulation changes from PPP to High-Level Data Link Control (HDLC), Cisco Discovery Protocol is reenabled on that interface even though it was explicitly disabled with the no cdp run command on that interface. This behavior is by design. The encapsulation changes the Layer 2 protocol configured for that interface and resets the interface configuration to the default Cisco Discovery Protocol state of being enabled, assuming that Cisco Discovery Protocol is enabled globally on the device.

Example: Disabling and Enabling Cisco Discovery Protocol on a Supported Interface

```
Device(config)# interface Gigabitethernet 1/0/1
Device(config-if)# [no] cdp enable
```

[**no**] **cdp enable**
Disables Cisco Discovery Protocol on the interface.

Note: If the encapsulation of an interface is changed, Cisco Discovery Protocol is reenabled on that interface even if Cisco Discovery Protocol was previously disabled.

Example: Setting the Transmission Timer and Hold Time

```
Device(config)# cdp timer 30
```

cdp timer *seconds*
Specifies the frequency of transmission of Cisco Discovery Protocol packets.

```
Device(config)# cdp holdtime 90
```

cdp holdtime *seconds*
Specifies the time for which a receiving device should hold information before discarding it.

Example: Disabling and Re-enabling Cisco Discovery Protocol Version 2 Advertisements

```
Device(config)# no cdp advertise-v2
```

[**no**] **cdp advertise-v2**
Disables the broadcasting of Cisco Discovery Protocol Version 2 advertisements.

Example: Monitoring and Maintaining Cisco Discovery Protocol

```
Device# clear cdp table
```

clear cdp table
Clears the table that contains Cisco Discovery Protocol information about neighbors.

```
Device# show cdp
```

show cdp
Displays the interval between advertisements, the number (in seconds) for which an advertisement is valid for a given port, and the version of the advertisement.

```
Device# show cdp interface
```

show cdp interface [*type number*]
Displays information about interfaces on which Cisco Discovery Protocol is enabled.

```
Device# show cdp neighbors
```

show cdp neighbors [*type number*] [**detail**]
Displays the type of device that has been discovered, the name of the device, the number and type of the local interface (port), the time (in seconds) the Cisco Discovery Protocol advertisement is valid for the interface, the device type, the device product number, and the port ID.
The **detail** keyword displays information about the native VLAN ID, the duplex mode, and the VTP domain name associated with neighboring devices.

```
Device# show cdp traffic
```

show cdp traffic
Displays information about Cisco Discovery Protocol traffic, including the number of packets sent and received and checksum errors.

LLDP

Reference: Configuring LLDP, LLDP-MED, and Wired Location Service

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the data link layer) on Cisco-manufactured devices. CDP allows network management applications to automatically discover and learn about other Cisco devices connected to the network.

To support non-Cisco devices and to allow for interoperability between other devices, the switch supports the IEEE 802.1AB Link Layer Discovery Protocol (LLDP). LLDP is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

Example: Enabling LLDP

```
Switch(config)# lldp run
```

lldp run
Enables LLDP globally on the switch.

```
Switch(config-if)# lldp transmit
```

lldp transmit
Enables the interface to send LLDP packets.

```
Switch(config-if)# lldp receive
```

lldp receive
Enables the interface to receive LLDP packets.

```
Switch# show lldp
```

show lldp
Verifies the configuration.

Example: Configuring LLDP Characteristics

```
Switch(config)# lldp holdtime 120
```

lldp holdtime *seconds*
Specifies the amount of time a receiving device should hold the information from your device before discarding it.
The range is 0 to 65535 seconds; the default is 120 seconds.

```
Switch(config)# lldp timer 30
```

lldp timer *rate*
Sets the sending frequency of LLDP updates in seconds.
The range is 5 to 65534 seconds; the default is 30 seconds.

UDLD

Reference:
Understanding and Configuring the Unidirectional Link Detection Protocol Feature
Layer 2 Configuration Guide, Cisco IOS Release 15.2(2)E (Catalyst 2960-X Switch)

Spanning-Tree Protocol (STP) resolves redundant physical topology into a loop-free, tree-like forwarding topology.

This is done by blocking one or more ports. By blocking one or more ports, there are no loops in the forwarding topology. STP relies on its operation on reception and transmission of the Bridge Protocol Data Units (BPDUs). If the STP process that runs on the switch with a blocking port stops receiving BPDUs from its upstream (designated) switch on the port, STP eventually ages out the STP information for the port and moves it to the forwarding state. This creates a forwarding loop or STP loop.

Packets start to cycle indefinitely along the looped path, and consume more and more bandwidth. This leads to a possible network outage.

How is it possible for the switch to stop receiving BPDUs while the port is up? The reason is unidirectional link. A link is considered unidirectional when the link is up on both sides of the connection but the local side is not receiving the packets sent by the remote side while the remote side receives packets sent by the local side.

In order to detect the unidirectional links before the forwarding loop is created, Cisco designed and implemented the UDLD protocol.

UDLD is a Layer 2 (L2) protocol that works with the Layer 1 (L1) mechanisms to determine the physical status of a link. At Layer 1, auto-negotiation takes care of physical signaling and fault detection. UDLD performs tasks that auto-negotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected ports. When you enable both auto-negotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

UDLD works by exchanging protocol packets between the neighboring devices. In order for UDLD to work, both devices on the link must support UDLD and have it enabled on respective ports.

Each switch port configured for UDLD sends UDLD protocol packets that contain the port's own device/port ID, and the neighbor's device/port IDs seen by UDLD on that port. Neighboring ports should see their own device/port ID (echo) in the packets received from the other side.

If the port does not see its own device/port ID in the incoming UDLD packets for a specific duration of time, the link is considered unidirectional.

This echo-algorithm allows detection of these issues:

- Link is up on both sides, however, packets are only received by one side.
- Wiring mistakes when receive and transmit fibers are not connected to the same port on the remote side.

Once the unidirectional link is detected by UDLD, the respective port is disabled and this message is printed on the console:

```
UDLD-3-DISABLE: Unidirectional link detected on port 1/2. Port
disabled
```

Port shutdown by UDLD remains disabled until it is manually re-enabled, or until errdisable timeout expires (if configured).
UDLD can operate in two modes: normal and aggressive.

In normal mode, if the link state of the port was determined to be bi-directional and the UDLD information times out, no action is taken by UDLD. The port state for UDLD is marked as undetermined. The port behaves according to its STP state.

In aggressive mode, if the link state of the port is determined to be bi-directional and the UDLD information times out while the link on the port is still up, UDLD tries to re-establish the state of the port. If not successful, the port is put into the errdisable state.

Aging of UDLD information happens when the port that runs UDLD does not receive UDLD packets from the neighbor port for the duration of hold time. The hold time for the port is dictated by the remote port and depends on the message interval at the remote side. This is 45 seconds for the default message interval of 15 seconds.

In aggressive mode, once the information is aged, UDLD will make an attempt to re-establish the link state by sending packets every second for eight seconds. If the link state is still not determined, the link is disabled.

Aggressive mode adds additional detection of these situations:

- The port is stuck (on one side the port neither transmits nor receives, however, the link is up on both sides).
- The link is up on one side and down on the other side. This issue might be seen on fiber ports. When transmit fiber is unplugged on the local port, the link remains up on the local side. However, it is down on the remote side.

Example: Configuration and Monitoring

```
Switch(config)# udld enable message time 10
```

udld {**aggressive** | **enable** | **message time** *message-timer-interval*}
Specifies the UDLD mode of operation:

- **aggressive**: Enables UDLD in aggressive mode on all fiber-optic ports.
- **enable**: Enables UDLD in normal mode on all fiber-optic ports on the switch. UDLD is disabled by default. An individual interface configuration overrides the setting of the **udld enable** global configuration command.
- **message time** *message-timer-interval*: Configures the period of time between UDLD probe messages on ports that are in the advertisement phase and are detected to be bidirectional. The range is from 1 to 90 seconds; the default value is 15.

Note: This command affects fiber-optic ports only. Use the udld interface configuration command to enable UDLD on other port types.
Use the **no** form of this command, to disable UDLD.

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

udld port [**aggressive**]
UDLD is disabled by default.

- **udld port**: Enables UDLD in normal mode on the specified port.
- **udld port aggressive**: (Optional) Enables UDLD in aggressive mode on the specified port.

Note: Use the **no udld port** interface configuration command to disable UDLD on a specified fiber-optic port.

VLAN Technologies

Access ports

Reference: LAN Switching Configuration Guide, Cisco IOS Release 15M&T

An access port can belong to one VLAN and is manually assigned to that VLAN.

Example: Configuring an Interface as Layer 2 Access

```
Device(config-if)# switchport mode access
```

switchport mode access
Configures the interface as a Layer 2 access.

```
Device(config-if)# switchport access vlan 10
```

switchport access vlan *vlan-number*
For access ports, specifies the access VLAN.

Trunk ports (802.1Q)

Reference:
LAN Switching Configuration Guide, Cisco IOS Release 15M&T

Configuring Access and Trunk Interfaces

A trunk port is a member of all VLANs by default, including extended-range VLANs, but membership can be limited by configuring the allowed-VLAN list. You can also modify the pruning-eligible list to block flooded traffic to VLANs on trunk ports that are included in the list.

IEEE 802.1Q is an industry-standard trunking encapsulation.

Ethernet trunk interfaces support different trunking modes. You can set an interface as trunking or non-trunking or to negotiate trunking with the neighboring interface. To autonegotiate trunking, the interfaces must be in the same VTP domain.

Trunk negotiation is managed by the Dynamic Trunking Protocol (DTP). However, some internetworking devices might forward DTP frames improperly, which could cause misconfigurations.

Example: Configuring an Interface as a Layer 2 Trunk

```
Device(config-if)# switchport mode trunk
```

switchport mode {**dynamic** {**auto** | **desirable**} | **trunk**}
Configures the interface as a Layer 2 trunk.
Note: Encapsulation is always dot1q.

switchport mode dynamic auto - Makes the interface able to convert the link to a trunk link. The interface becomes a trunk interface if the neighboring interface is set to **trunk** or **desirable** mode.

switchport mode dynamic desirable - Makes the interface actively attempt to convert the link to a trunk link. The interface becomes a trunk interface if the neighboring interface is set to **trunk**, **desirable**, or **auto** mode.

switchport mode trunk - Puts the interface into permanent trunking mode and negotiates to convert the neighboring link into a trunk link. The interface becomes a trunk interface even if the neighboring interface is not a trunk interface.

switchport nonegotiate - Prevents the interface from generating DTP frames. You can use this command only when the interface switchport mode is **access** or **trunk**. You must manually configure the neighboring interface as a trunk interface to establish a trunk link.

Native VLAN

Reference: Configuring VLAN Trunks

A trunk port configured with IEEE 802.1Q tagging can receive both tagged and untagged traffic. By default, the switch forwards untagged traffic in the native VLAN configured for the port. The native VLAN is VLAN 1 by default.

The native VLAN can be assigned any VLAN ID.

If a packet has a VLAN ID that is the same as the outgoing port native VLAN ID, the packet is sent untagged; otherwise, the switch sends the packet with a tag.

When you connect two Cisco switches through 802.1Q trunks, the switches exchange spanning-tree bridge protocol data units (BPDUs) on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1d spanning-tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Shared Spanning Tree Protocol (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).

Example: Configuring the Native VLAN for Untagged Traffic

```
Switch(config-if)# switchport trunk native vlan 12
```

switchport trunk native vlan *vlan-id*
Configures the VLAN that is sending and receiving untagged traffic on the trunk port.

For *vlan-id*, the range is 1 to 4094.

Manual VLAN pruning

Reference:
Cisco IOS Interface and Hardware Component Command Reference
Configuring VLAN Trunks

By default, a trunk port sends traffic to and receives traffic from all VLANs. All VLAN IDs, 1 to 4094, are allowed on each trunk. However, you can remove VLANs from the allowed list, preventing traffic from those VLANs from passing over the trunk.

VLAN 1 is the default VLAN on all trunk ports in all Cisco switches, and it has previously been a requirement that VLAN 1 always be enabled on every trunk link. When you remove VLAN 1 from a trunk port, the interface continues to send and receive management traffic, for example, Cisco Discovery Protocol (CDP), Port Aggregation Protocol (PAgP), Link Aggregation Control Protocol (LACP), DTP, and VTP in VLAN 1. No user traffic is sent or received on VLAN 1.

Example: Defining the Allowed VLANs on a Trunk

```
Device(config-if)# switchport trunk allowed vlan
1-2,40,60,1002-1005
```

switchport trunk allowed vlan { *vlan-id* | **add** | **all** | **except** | **none** | **remove**} *vlan-list*
Configures the list of VLANs allowed on the trunk.

The *vlan-list* parameter is either a single VLAN number from 1 to 4094 or a range of VLANs described by two VLAN numbers, the lower one first, separated by a hyphen. Do not enter any spaces between comma-separated VLAN parameters or in hyphen-specified ranges.

add -- Adds the defined list of VLANs to those currently set instead of replacing the list.

all -- Specifies all VLANs from 1 to 1005. Beginning with Cisco IOS Release 12.4(15)T, the valid VLAN ID range is from 1 to 4094.

except -- Lists the VLANs that should be calculated by inverting the defined list of VLANs.

none -- Indicates an empty list. This keyword is not supported in the **switchport trunk allowed vlan** form of the command.

remove -- Removes the defined list of VLANs from those currently set instead of replacing the list.

VLAN database

Reference:
VLAN Configuration Guide
Cisco IOS LAN Switching Command Reference

VLANs are divided into two groups on Cisco devices: normal range VLANs (1-1005) and extended range VLANs (1006-4094).

Normal range VLANs are always stored in the vlan.dat file, and in addition, if the switch operates in VTP Transparent mode, they also appear in the running-config.

Extended range VLANs are always stored in the running-configuration, and if VTPv3 is used, also in the vlan.dat file (not with older VTP versions).

To add a VLAN and enter config-VLAN submode, use the **vlan** command in global configuration mode. To delete the VLAN, use the **no** form of this command.

```
Device(config)# vlan 2

Device(config)# vlan 2,5,10-12,20,25,4000
```

vlan {*vlan-id* | *vlan-range*}

no vlan {*vlan-id* | *vlan-range*}

vlan-id - Number of the VLAN; valid values are from 1 to 4094.
vlan-range - Range of configured VLANs.

Normal range and extended range VLANs

Reference: VLAN Configuration Guide

Normal-range VLANs are VLANs with IDs from 1 to 1005.

Normal-range VLANs are identified with a number between 1 and 1001. VLAN numbers 1002 through 1005 are reserved for Token Ring and FDDI VLANs.

VLAN configurations for VLANs 1 to 1005 are always saved in the VLAN database. If the VTP mode is transparent, VTP and VLAN configurations are also saved in the switch running configuration file.

If the switch is in VTP server or VTP transparent mode, you can add, modify or remove configurations for VLANs 2 to 1001 in the VLAN database. (VLAN IDs 1 and 1002 to 1005 are automatically created and cannot be removed.)

With VTP versions 1 and 2, the switch supports VLAN IDs 1006 through 4094 only in VTP transparent mode (VTP disabled). These are extended-range VLANs and configuration options are limited. Extended-range VLANs created in VTP transparent mode are not saved in the VLAN database and are not propagated. VTP version 3 supports extended range VLAN (VLANs 1006 to 4094) database propagation in VTP server mode. If extended VLANs are configured, you cannot convert from VTP version 3 to version 1 or 2.

Extended-range VLANs are VLANs with IDs from 1006 to 4094.

VTP 3 only supports extended-range VLANs.

VLAN IDs in the extended range are not saved in the VLAN database and are not recognized by VTP unless the switch is running VTP version 3.

VTP version 1 and 2, a switch must be in VTP transparent mode when you create extended-range VLANs. If VTP mode is server or client, an error message is generated, and the extended-range VLAN is rejected. VTP version 3 supports extended VLANs in server and transparent modes.

configuration to the startup configuration so that the switch boots up in VTP transparent mode. Otherwise, you lose the extended-range VLAN configuration if the switch resets. If you create extended-range VLANs in VTP version 3, you cannot convert to VTP version 1 or 2.

Voice VLAN

Reference:
VLAN Configuration Guide
Cisco IOS Interface and Hardware Component Command Reference

A voice VLAN port is an access port attached to a Cisco IP Phone, configured to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the phone.

With the voice VLAN feature, network administrators can segment phones into separate logical networks, even though the data and voice infrastructure is physically the same. The voice VLAN feature places the phones into their own VLANs without the need for end-user intervention.

User priority bits in the 802.1p portion of the 802.1Q standard header are used to provide prioritization in Ethernet devices. This is a vital component in designing Cisco AVVID networks.

```
Device(config-if)# switchport mode access
Device(config-if)# switchport access vlan 100
Device(config-if)# switchport voice vlan 101
```

switchport voice vlan {*vlan-id* | **dot1p** | **none** | **untagged**}
Configures the voice port with a VVID that will be used exclusively for voice traffic.

vlan-id - Voice VLAN identifier (VVID) of the VLAN used for voice traffic. Valid IDs are from 1 to 1005 (IDs 1006 to 4096 are not supported).

dot1p - The telephone uses priority tagging and uses VLAN 0. The switch port is an 802.1Q trunk port.

none - The telephone is not instructed through the command line interface (CLI) about the voice VLAN. The telephone uses its own configuration from the telephone keypad and transmits untagged voice traffic in the default VLAN.

untagged - The telephone does not tag frames; it uses VLAN 4095. The switch port can be an access port or an 802.1Q trunk port.

VTP

Reference:
Understanding and Configuring VTP
Configuring VTP

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs within a VTP domain. A VTP domain is made up of network devices that share the same VTP domain name and that are interconnected with trunks. With VTP, you can make configuration changes centrally on one or more network devices and have those changes automatically communicated to all the other network devices in the network.

A network device can be configured to be in only one VTP domain.

If the switch receives a VTP advertisement over a trunk link, it inherits the management domain name and the VTP configuration revision number. The switch ignores advertisements with a different management domain name or an earlier configuration revision number.

A switch can operate in any one of these VTP modes:

- Server — In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) for the entire VTP domain. VTP servers advertise their VLAN configuration to other network devices in the same VTP domain and synchronize their VLAN configuration with other network devices based on advertisements received over trunk links. VTP server is the default mode.
- Client — VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- Transparent — VTP transparent network devices do not participate in VTP. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements.

VTP version 2 supports the following features, which are not supported in version 1:
Version-Dependent Transparent Mode—In VTP version 1, a VTP transparent network device inspects VTP messages for the domain name and version, and forwards a message only if

the version and domain name match. Because only one domain is supported in the supervisor engine software, VTP version 2 forwards VTP messages in transparent mode, without checking the version.

VTP pruning enhances network bandwidth use by reducing unnecessary flooded traffic, such as broadcast, multicast, and unicast packets. VTP pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices. By default, VTP pruning is disabled.

For VTP pruning to be effective, all devices in the management domain must either support VTP pruning or, on devices that do not support VTP pruning, you must manually configure the VLANs allowed on trunks.

Enabling VTP pruning on a VTP server enables pruning for the entire management domain. VTP pruning takes effect several seconds after you enable it. By default, VLANs 2 through 1000 are eligible for pruning. VTP pruning does not prune traffic from pruning-ineligible VLANs. VLAN 1 is always ineligible for pruning; traffic from VLAN 1 cannot be pruned.

Example: Configuring VTP

```
switch(config)# vtp domain CCIE_EI
```

vtp domain *domain-name*
Specifies the name of the VTP domain that you want this device to join. The default is blank.

```
switch(config)# vtp mode transparent
```

vtp mode {**client** | **server** | **transparent** | **off**}
Sets the VTP mode to client, server, transparent, or off.

```
switch(config)# vtp password cisco
```

vtp password *password*
Specifies the password for the VTP administrative domain.

```
switch(config)# vtp version 2
```

vtp version {**1** | **2**}
Sets the VTP version that you want to use. The default is version 1.

Example: Configuring VTP Pruning

EtherChannel

Reference: Software Configuration Guide, Cisco IOS Release 15.2(4)E (Catalyst 3750-X and 3560-X Switches)

EtherChannel provides fault-tolerant, high-speed links between switches, routers, and servers. EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links. If a link fails, EtherChannel redirects traffic from the failed link to the remaining links in the channel without intervention.

All ports in each EtherChannel must be configured as either Layer 2 or Layer 3 ports.

You can configure an EtherChannel in one of these modes: Port Aggregation Protocol (PAgP), Link Aggregation Control Protocol (LACP), or On. Configure both ends of the EtherChannel in the same mode:

- When you configure one end of an EtherChannel in either PAgP or LACP mode, the system negotiates with the other end of the channel to determine which ports should become active. If the remote port cannot negotiate an EtherChannel, the local port is put into an independent state and continues to carry data traffic as would any other single link. The port configuration does not change, but the port does not participate in the EtherChannel.
- When you configure an EtherChannel in the on mode, no negotiations take place. The switch forces all compatible ports to become active in the EtherChannel. The other end of the channel (on the other switch) must also be configured in the on mode; otherwise, packet loss can occur.

The **channel-group** command binds the physical port and the port-channel interface together. Each EtherChannel has a port-channel logical interface numbered from 1 to 48. This port-channel interface number corresponds to the one specified with the **channel-group** interface configuration command.

With Layer 2 ports, use the **channel-group** interface configuration command to dynamically create the port-channel interface.

You also can use the **interface port-channel** *port-channel-number* global configuration command to manually create the port-channel interface, but then you must use the **channel-group** *channel-group-number* command to bind the logical interface to a physical port. The *channel-group-number* can be the same as the *port-channel-number*, or you can use a new number. If you use a new number, the **channel-group** command dynamically creates a new port channel.

With Layer 3 ports, you should manually create the logical interface by using the **interface port-channel** global configuration command followed by the **no switchport** interface configuration command. You then manually assign an interface to the EtherChannel by using the **channel-group** interface configuration command.

In Layer 2 EtherChannels, the first port in the channel that comes up provides its MAC address to the EtherChannel. If this port is removed from the bundle, one of the remaining ports in the bundle provides its MAC address to the EtherChannel.

For Layer 3 EtherChannels, the MAC address is allocated by the active switch as soon as the interface is created through the **interface port-channel** global configuration command.

LACP, static

Link Aggregation Control Protocol (LACP)

The LACP is defined in IEEE 802.3ad and enables Cisco switches to manage Ethernet channels between switches that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between Ethernet ports.

LACP modes specify whether a port can send LACP packets or only receive LACP packets.

active - Places a port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets.

passive - Places a port into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation. This setting minimizes the transmission of LACP packets.

Ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible.

- A port in the active mode can form an EtherChannel with another port that is in the active or passive mode.
- A port in the passive mode cannot form an EtherChannel with another port that is also in the passive mode because neither port starts LACP negotiation.

Static (EtherChannel On Mode)

EtherChannel on mode can be used to manually configure an EtherChannel. The on mode forces a port to join an EtherChannel without negotiations. The on mode can be useful if the remote device does not support PAgP or LACP. In the on mode, a usable EtherChannel exists only when the switches at both ends of the link are configured in the on mode.

Ports that are configured in the on mode in the same channel group must have compatible port characteristics, such as speed and duplex. Ports that are not compatible are suspended, even though they are configured in the on mode.

NOTE: You should use care when using the on mode. This is a manual configuration, and ports on both ends of the EtherChannel must have the same configuration. If the group is misconfigured, packet loss or spanning-tree loops can occur.

Layer 2, Layer 3

The EtherChannel Layer 3 ports are made up of routed ports. Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command.

Layer 2

You configure Layer 2 EtherChannels by assigning ports to a channel group with the **channel-group** interface configuration command. This command automatically creates the port-channel logical interface.

Example: Configuring Layer 2 EtherChannels

```
Switch(config)# interface gigabitethernet2/0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 22
Switch(config-if)# channel-group 5 mode active
```

channel-group *channel-group-number* **mode** {**active** | **passive**}
Assigns the port to a channel group, and specifies the LACP mode.

active — Enables LACP only if a LACP device is detected. It places the port into an active

negotiating state in which the port starts negotiations with other ports by sending LACP packets.
passive — Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.

channel-group *channel-group-number* **mode on**
on — Forces the port to channel without PAgP or LACP. In the **on** mode, an EtherChannel exists only when a port group in the **on** mode is connected to another port group in the **on** mode.

Layer 3

When configuring Layer 3 EtherChannels, you should first manually create the port-channel logical interface by using the **interface port-channel** global configuration command. Then put the logical interface into the channel group by using the **channel-group** interface configuration command.

NOTE: To move an IP address from a physical port to an EtherChannel, you must delete the IP address from the physical port before configuring it on the port-channel interface.

Example: Configuring Layer 3 EtherChannels

```
Switch(config)# interface port-channel 5
Switch(config-if)# no switchport
Switch(config-if)# ip address ip address 172.10.20.10
255.255.255.0
```

interface port-channel *port-channel-number*
Specifies the port-channel logical interface, and enters interface configuration mode. For *port-channel-number*, the range is 1 to 48.

no switchport
Puts the interface into Layer 3 mode.

ip address *ip-address mask*
Assigns an IP address and subnet mask to the EtherChannel.

```
Switch(config)# interface gigabitethernet2/0/1
Switch(config-if)# no ip address
Switch(config-if)# no switchport
Switch(config-if)# channel-group 5 mode active
```

no ip address
Ensures that there is no IP address assigned to the physical port.

no switchport
Puts the port into Layer 3 mode.

channel-group *channel-group-number* **mode** {**active** | **passive**}
Assigns the port to a channel group, and specifies the LACP mode.

active — Enables LACP only if a LACP device is detected. It places the port into an active negotiating state in which the port starts negotiations with other ports by sending LACP packets.
passive — Enables LACP on the port and places it into a passive negotiating state in which the port responds to LACP packets that it receives, but does not start LACP packet negotiation.

channel-group *channel-group-number* **mode on**
on — Forces the port to channel without PAgP or LACP. In the **on** mode, an EtherChannel exists only when a port group in the **on** mode is connected to another port group in the **on** mode.

Load balancing

Reference: Understanding EtherChannel Load Balancing and Redundancy on Catalyst Switches

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. You can specify one of several different load-balancing modes, including load distribution based on MAC addresses, IP addresses, source addresses, destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch.

Example: Configuring EtherChannel Load Balancing

```
Switch(config)# port-channel load-balance src-mac
```

port-channel load-balance {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}
Configures an EtherChannel load-balancing method. The default is src-mac.

- **dst-ip** — Specifies destination-host IP address.
- **dst-mac** — Specifies the destination-host MAC address of the incoming packet.
- **src-dst-ip** — Specifies the source and destination host IP address.
- **src-dst-mac** — Specifies the source and destination host MAC address.
- **src-ip** — Specifies the source host IP address.
- **src-mac** — Specifies the source MAC address of the incoming packet.

Configuring LACP Hot-Standby Ports

When LACP is enabled, the software, by default, tries to configure the maximum number of LACP-compatible ports in a channel, up to a maximum of 16 ports. Only eight LACP links can be active at one time; the remaining eight links are placed in hot-standby mode. If one of the active links becomes inactive, a link that is in the hot-standby mode becomes active in its place.

You can override the default behavior by specifying the maximum number of active ports in a channel, in which case, the remaining ports become hot-standby ports.

If you configure more than eight links for an EtherChannel group, the software automatically decides which of the hot-standby ports to make active based on the LACP priority. To every link between systems that operate LACP, the software assigns a unique priority made up of these elements (in priority order):

- LACP system priority
- System ID (the switch MAC address)
- LACP port priority
- Port number

In priority comparisons, numerically lower values have higher priority. The priority decides which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating.

You can configure the system priority for all the EtherChannels that are enabled for LACP by using the **lacp system-priority** global configuration command.

Example: Configuring the LACP System Priority

```
Switch(config)# lacp system-priority 32000
```

lacp system-priority *priority*
Configures the LACP system priority.
The range is 1 to 65535. The default is 32768.
The lower the value, the higher the system priority.

By default, all ports use the same port priority. If the local system has a lower value for the system priority and the system ID than the remote system, you can affect which of the hot-standby links become active first by changing the port priority of LACP EtherChannel ports to a lower value than the default. The hot-standby ports that have lower port numbers become active in the channel first.

Example: Configuring the LACP Port Priority

```
Switch(config)# interface gigabitethernet 1/0/2
Switch(config-if)# lacp port-priority 32000
```

lacp port-priority *priority*
Configures the LACP port priority.
The range is 1 to 65535. The default is 32768.
The lower the value, the more likely that the port will be used for LACP transmission.

You can specify the minimum number of active ports that must be in the link-up state and bundled in an EtherChannel for the port channel interface to transition to the link-up state. Using EtherChannel **min-links**, you can prevent low-bandwidth LACP EtherChannels from becoming active. Port channel **min-links** also cause LACP EtherChannels to become inactive if they have too few active member ports to supply the required minimum bandwidth.

Example: Configuring the LACP Port Channel Min-Links Feature

```
Switch(config)# interface port-channel 2
Switch(config-if)# port-channel min-links 3
```

port-channel min-links *min-links-number*
Specifies the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state.
For *min-links-number*, the range is 2 to 8.

EtherChannel Misconfiguration Guard

Reference: Configuring Optional Spanning-Tree Features

You can use EtherChannel guard to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the switch interfaces are configured in an EtherChannel, but the interfaces on the other device are not. A misconfiguration can also occur if the channel parameters are not the same at both ends of the EtherChannel.

If the switch detects a misconfiguration on the other device, EtherChannel guard places the switch interfaces in the error-disabled state, and displays an error message.

You can enable this feature by using the spanning-tree etherchannel guard misconfig global configuration command.

Example: Enabling EtherChannel Guard

```
Switch(config)# spanning-tree etherchannel guard misconfig
```

spanning-tree etherchannel guard misconfig
Enable EtherChannel guard.

Spanning-Tree Protocol

PVST+, Rapid PVST+, MST

Reference: Configuring Spanning Tree Protocol

The stable, active spanning-tree topology of a switched network is controlled by these elements:

- The unique bridge ID (switch priority and MAC address) associated with each VLAN on each switch. In a switch stack, all switches use the same bridge ID for a given spanning-tree instance.
- The spanning-tree path cost to the root switch.
- The port identifier (port priority and MAC address) associated with each Layer 2 interface.

The BPDUs communicate and compute the spanning-tree topology. Each configuration BPDU contains this information:

- The unique bridge ID of the switch that the sending switch identifies as the root switch
- The spanning-tree path cost to the root
- The bridge ID of the sending switch
- Message age
- The identifier of the sending interface
- Values for the hello, forward delay, and max-age protocol timers

When a switch receives a configuration BPDU that contains superior information (lower bridge ID, lower path cost, and so forth), it stores the information for that port. If this BPDU is received on the root port of the switch, the switch also forwards it with an updated message to all attached LANs for which it is the designated switch.

If a switch receives a configuration BPDU that contains inferior information to that currently stored for that port, it discards the BPDU. If the switch is a designated switch for the LAN from which the inferior BPDU was received, it sends that LAN a BPDU containing the up-to-date information stored for that port. In this way, inferior information is discarded, and superior information is propagated on the network.

A BPDU exchange results in these actions:

- One switch in the network is elected as the root switch (the logical center of the spanning-tree topology in a switched network). For each VLAN, the switch with the highest switch priority (the lowest numerical priority value) is elected as the root switch. If all switches are configured with the default priority (32768), the switch with the lowest MAC address in the VLAN becomes the root switch. The switch priority value occupies the most significant bits of the bridge ID, as shown in the following figure.
- A root port is selected for each switch (except the root switch). This port provides the best path (lowest cost) when the switch forwards packets to the root switch.
- When selecting the root port on a switch stack, spanning tree follows this sequence:
  - Selects the lowest root bridge ID
  - Selects the lowest path cost to the root switch
  - Selects the lowest designated bridge ID
  - Selects the lowest designated path cost
  - Selects the lowest port ID
- The shortest distance to the root switch is calculated for each switch based on the path cost.
- A designated switch for each LAN segment is selected. The designated switch incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated switch is attached to the LAN is called the designated port.

When the spanning-tree algorithm places a Layer 2 interface in the forwarding state, this process occurs:

1. The interface is in the listening state while spanning tree waits for protocol information to move the interface to the blocking state.
2. While spanning tree waits for the forward-delay timer to expire, it moves the interface to the learning state and resets the forward-delay timer.
3. In the learning state, the interface continues to block frame forwarding as the switch learns end-station location information for the forwarding database.
4. When the forward-delay timer expires, spanning tree moves the interface to the forwarding state, where both learning and frame forwarding are enabled.

The switch supports these spanning-tree modes and protocols:

- PVST+ — This spanning-tree mode is based on the IEEE 802.1D standard and Cisco proprietary extensions. The PVST+ runs on each VLAN on the switch up to the maximum supported, ensuring that each has a loop-free path through the network.
- Rapid PVST+—This spanning-tree mode is the same as PVST+ except that is uses a rapid convergence based on the IEEE 802.1w standard. Beginning from the 15.2(4)E release, the default mode of STP is Rapid PVST+ . To provide rapid convergence, the Rapid PVST+ immediately deletes dynamically learned MAC address entries on a per-port basis upon receiving a topology change. By contrast, PVST+ uses a short aging time for dynamically learned MAC address entries.
- MSTP—This spanning-tree mode is based on the IEEE 802.1s standard. You can map multiple VLANs to the same spanning-tree instance, which reduces the number

of spanning-tree instances required to support a large number of VLANs. The MSTP runs on top of the RSTP (based on IEEE 802.1w), which provides for rapid convergence of the spanning tree by eliminating the forward delay and by quickly transitioning root ports and designated ports to the forwarding state. In a switch stack, the cross-stack rapid transition (CSRT) feature performs the same function as RSTP. You cannot run MSTP without RSTP or CSRT.

In a mixed MSTP and PVST+ network, the common spanning-tree (CST) root must be inside the MST backbone, and a PVST+ switch cannot connect to multiple MST regions.

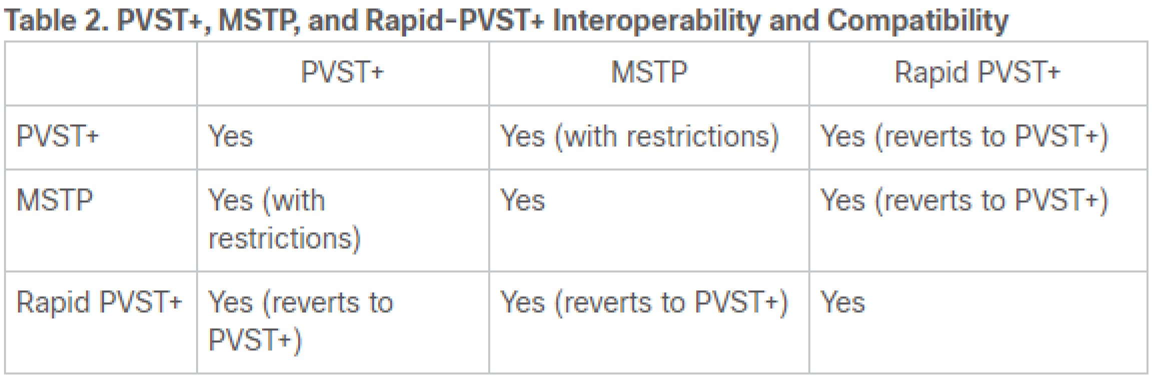


Table 2. PVST+, MSTP, and Rapid-PVST+ Interoperability and Compatibility

	PVST+	MSTP	Rapid PVST+
PVST+	Yes	Yes (with restrictions)	Yes (reverts to PVST+)
MSTP	Yes (with restrictions)	Yes	Yes (reverts to PVST+)
Rapid PVST+	Yes (reverts to PVST+)	Yes (reverts to PVST+)	Yes

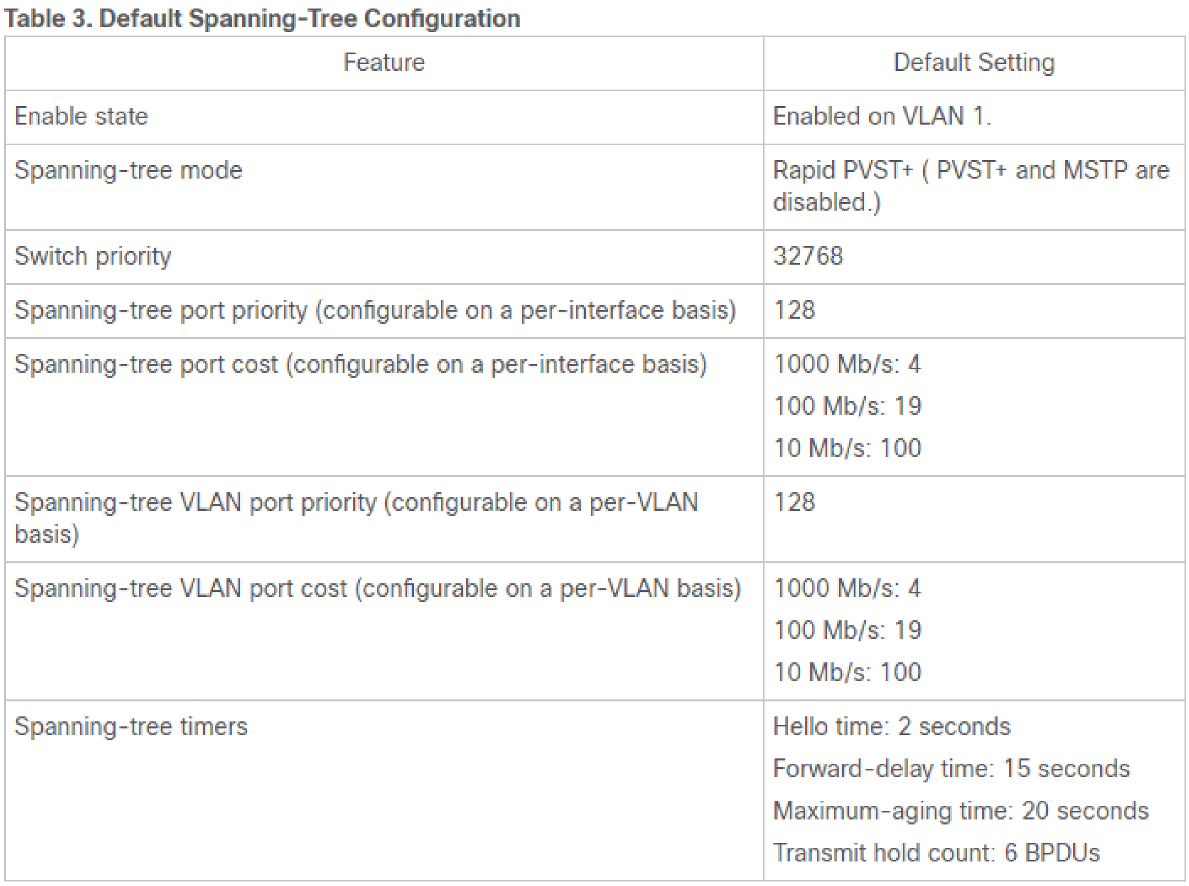


Table 3. Default Spanning-Tree Configuration

Feature	Default Setting
Enable state	Enabled on VLAN 1.
Spanning-tree mode	Rapid PVST+ (PVST+ and MSTP are disabled.)
Switch priority	32768
Spanning-tree port priority (configurable on a per-interface basis)	128
Spanning-tree port cost (configurable on a per-interface basis)	1000 Mb/s: 4 100 Mb/s: 19 10 Mb/s: 100
Spanning-tree VLAN port priority (configurable on a per-VLAN basis)	128
Spanning-tree VLAN port cost (configurable on a per-VLAN basis)	1000 Mb/s: 4 100 Mb/s: 19 10 Mb/s: 100
Spanning-tree timers	Hello time: 2 seconds Forward-delay time: 15 seconds Maximum-aging time: 20 seconds Transmit hold count: 6 BPDUs

If more VLANs are defined in the VTP than there are spanning-tree instances, you can enable PVST+ or rapid PVST+ on only 128 VLANs on the switch or each switch stack. The remaining VLANs operate with spanning tree disabled. However, you can map multiple VLANs to the same spanning-tree instances by using MSTP.

If 128 instances of spanning tree are already in use, you can disable spanning tree on one of the VLANs and then enable it on the VLAN where you want it to run. Use the no spanning-tree vlan vlan-id global configuration command to disable spanning tree on a specific VLAN, and use the spanning-tree vlan vlan-id global configuration command to enable spanning tree on the desired VLAN.

The switch supports PVST+, rapid PVST+, and MSTP, but only one version can be active at any time. (For example, all VLANs run PVST+, all VLANs run rapid PVST+, or all VLANs run MSTP.)

Example: Changing the Spanning-Tree Mode

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

spanning-tree mode {**pvst** | **mst** | **rapid-pvst**}
Configures a spanning-tree mode. All stack members run the same version of spanning tree.

- Select **pvst** to enable PVST+.
- Select **mst** to enable MSTP.
- Select **rapid-pvst** to enable rapid PVST+.

Spanning tree is enabled by default on VLAN 1 and on all newly created VLANs up to the spanning-tree limit. Disable spanning tree only if you are sure there are no loops in the network topology.

WARNING: When spanning tree is disabled and loops are present in the topology, excessive traffic and indefinite packet duplication can drastically reduce network performance.

Example: Disabling Spanning-Tree

```
Switch(config)# no spanning-tree vlan 300
```

no spanning-tree vlan *vlan-id*
For *vlan-id*, the range is 1 to 4094.

Multiple Spanning-Tree Protocol

Reference: Configuring Multiple Spanning-Tree Protocol

When you enable MST by using the spanning-tree mode mst global configuration command, RSTP is automatically enabled.

For two or more switches to be in the same multiple spanning tree (MST) region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same name.

The switch stack supports up to 65 MST instances. The number of VLANs that can be mapped to a particular MST instance is unlimited.

For switches to participate in multiple spanning-tree (MST) instances, you must consistently configure the switches with the same MST configuration information. A collection of interconnected switches that have the same MST configuration comprises an MST region.

The MST configuration controls to which MST region each switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map. You configure the switch for a region by specifying the MST region configuration on it. You can map VLANs to an MST instance, specify the region name, and set the revision number.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing RSTP bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network, but each region can support up to 65 spanning-tree instances. Instances can be identified by any number in the range from 0 to 4094. You can assign a VLAN to only one spanning-tree instance at a time.

Unlike PVST+ and Rapid PVST+ in which all the spanning-tree instances are independent, the MSTP establishes and maintains two types of spanning trees: 1) an internal spanning tree (IST) and 2) a common and internal spanning tree (CIST)

An internal spanning tree (IST), which is the spanning tree that runs in an MST region.

Within each MST region, the MSTP maintains multiple spanning-tree instances. Instance 0 is a special instance for a region, known as the internal spanning tree (IST). All other MST instances are numbered from 1 to 4094.

The IST is the only spanning-tree instance that sends and receives BPDUs. All of the other spanning-tree instance information is contained in M-records, which are encapsulated within MSTP BPDUs. Because the MSTP BPDU carries information for all instances, the number of BPDUs that need to be processed to support multiple spanning-tree instances is significantly reduced.

All MST instances within the same region share the same protocol timers, but each MST instance has its own topology parameters, such as root switch ID, root path cost, and so forth. By default, all VLANs are assigned to the IST.

An MST instance is local to the region; for example, MST instance 1 in region A is independent of MST instance 1 in region B, even if regions A and B are interconnected.

A common and internal spanning tree (CIST), which is a collection of the ISTs in each MST region, and the common spanning tree (CST) that interconnects the MST regions and single spanning trees.

The spanning tree computed in a region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning-tree algorithm running among switches that support the IEEE 802.1w, IEEE 802.1s, and IEEE 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Switch priority, port priority, path cost, STP timers

Reference: Configuring Spanning Tree Protocol

Switch Priority

To configure a switch as the root for the specified VLAN, use the **spanning-tree vlan** *vlan-id* root global configuration command to modify the switch priority from the default value (32768) to a significantly lower value.

Use the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of switch hops between any two end stations in the Layer 2 network). When you specify the network diameter, the switch automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time.

Example: Configuring the Root Switch

```
Switch(config)# spanning-tree vlan 20-24 root primary
```

spanning-tree vlan *vlan-id* **root primary** [**diameter** *net-diameter* [**hello-time** *seconds*]]
Configures a switch to become the root for the specified VLAN.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- (Optional) For diameter *net-diameter*, specify the maximum number of switches between any two end stations. The range is 2 to 7.
- (Optional) For hello-time *seconds*, specify the interval in seconds between the generation of configuration messages by the root switch. The range is 1 to 10; the default is 2.

After configuring the switch as the root switch, we recommend that you avoid manually configuring the hello time, forward-delay time, and maximum-age time through the **spanning-tree vlan** *vlan-id* **hello-time**, **spanning-tree vlan** *vlan-id* **forward-time**, and the **spanning-tree vlan** *vlan-id* **max-age** global configuration commands.

When you configure a switch as the secondary root, the switch priority is modified from the default value (32768) to 28672. With this priority, the switch is likely to become the root switch for the specified VLAN if the primary root switch fails. This is assuming that the other network switches use the default switch priority of 32768, and therefore, are unlikely to become the root switch.

Example: Configuring a Secondary Root Switch

```
Switch(config)# spanning-tree vlan 20-24 root secondary
```

spanning-tree vlan *vlan-id* **root secondary** [**diameter** *net-diameter* [**hello-time** *seconds*]]
Configures a switch to become the secondary root for the specified VLAN.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- (Optional) For diameter *net-diameter*, specify the maximum number of switches between any two end stations. The range is 2 to 7.
- (Optional) For hello-time *seconds*, specify the interval in seconds between the generation of configuration messages by the root switch. The range is 1 to 10; the default is 2.

Port Priority

If a loop occurs, spanning tree uses the port priority when selecting an interface to put into the forwarding state. You can assign higher priority values (lower numerical values) to interfaces that you want to select first and lower priority values (higher numerical values) that you want to select last. If all interfaces have the same priority value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Example: Configuring Port Priority

```
Switch(config-if)# spanning-tree port-priority 0
```

spanning-tree port-priority *priority*
Configures the port priority for an interface.

- For *priority*, the range is 0 to 240, in increments of 16; the default is 128. Valid values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected. The lower the number, the higher the priority.

```
Switch(config-if)# spanning-tree vlan 20-25 port-priority 0
```

spanning-tree vlan *vlan-id* **port-priority** *priority*
Configures the port priority for a VLAN.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- For *priority*, the range is 0 to 240, in increments of 16; the default is 128. Valid values are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240. All other values are rejected. The lower the number, the higher the priority.

Path Cost

The spanning-tree path cost default value is derived from the media speed of an interface. If a loop occurs, spanning tree uses cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want to select first and higher cost values that you want to select last. If all interfaces have the same cost value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Example: Configuring Path Cost

```
Switch(config-if)# spanning-tree cost 250
```

spanning-tree cost *cost*
Configures the cost for an interface.

- For *cost*, the range is 1 to 200000000; the default value is derived from the media speed of the interface.

```
Switch(config-if)# spanning-tree vlan 10,12-15,20 cost 300
```

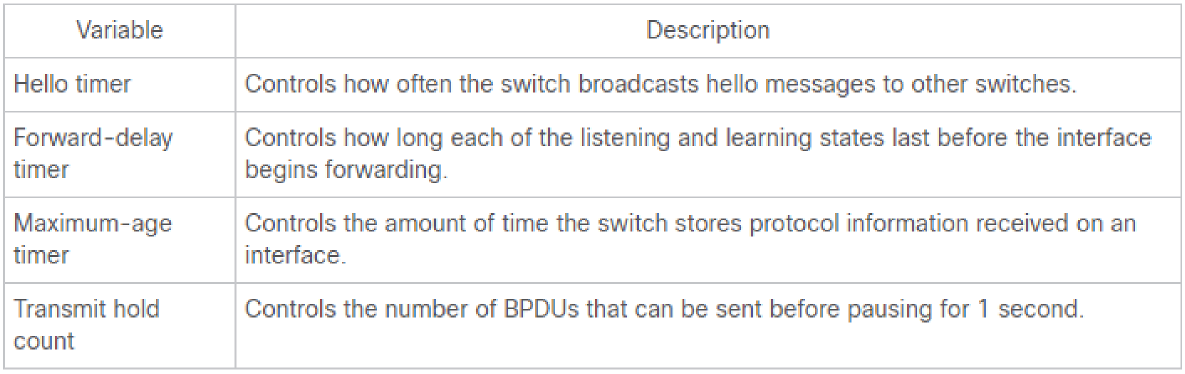
spanning-tree vlan *vlan-id* **cost** *cost*
Configures the cost for a VLAN.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- For *cost*, the range is 1 to 200000000; the default value is derived from the media speed of the interface.

STP Timers

These timers affect the entire spanning-tree performance.

Table 4. Spanning-Tree Timers



Variable	Description
Hello timer	Controls how often the switch broadcasts hello messages to other switches.
Forward-delay timer	Controls how long each of the listening and learning states last before the interface begins forwarding.
Maximum-age timer	Controls the amount of time the switch stores protocol information received on an interface.
Transmit hold count	Controls the number of BPDUs that can be sent before pausing for 1 second.

Example: Configuring Spanning Tree Timers

```
Switch(config)# spanning-tree vlan 20-24 hello-time 3
```

spanning-tree vlan *vlan-id* **hello-time** *seconds*
Configures the hello time of a VLAN. The hello time is the time interval between configuration messages generated and sent by the root switch. These messages mean that the switch is alive.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- For *seconds*, the range is 1 to 10; the default is 2.

```
Switch(config)# spanning-tree vlan 20,25 forward-time 18
```

spanning-tree vlan *vlan-id* **forward-time** *seconds*
Configures the forward time of a VLAN. The forwarding delay is the number of seconds an interface waits before changing from its spanning-tree learning and listening states to the forwarding state.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- For *seconds*, the range is 4 to 30; the default is 15.

```
Switch(config)# spanning-tree vlan 20 max-age 30
```

spanning-tree vlan *vlan-id* **max-age** *seconds*
Configures the maximum aging time of a VLAN. The maximum-aging time is the number of seconds a switch waits without receiving spanning-tree configuration messages before attempting a reconfiguration.

- For *vlan-id*, you can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.
- For *seconds*, the range is 6 to 40; the default is 20.

```
Switch(config)# spanning-tree transmit hold-count 6
```

spanning-tree transmit hold-count *value*
Configures the number of BPDUs that can be sent before pausing for 1 second.

- For *value*, the range is 1 to 20; the default is 6.

PortFast, BPDU Guard, BPDU Filter

Reference: Configuring Optional Spanning-Tree Features

PortFast

PortFast immediately brings an interface configured as an access or trunk port to the forwarding state from a blocking state, bypassing the listening and learning states.

PortFast minimizes the time that interfaces must wait for spanning tree to converge, so it is effective only when used on interfaces connected to end stations. If you enable PortFast on an interface connecting to another switch, you risk creating a spanning-tree loop.

You can enable this feature by enabling it on either the interface or on all non-trunking ports.

An interface with the PortFast feature enabled is moved directly to the spanning-tree forwarding state without waiting for the standard forward-time delay.

If you enable the voice VLAN feature, the PortFast feature is automatically enabled. When you disable voice VLAN, the PortFast feature is not automatically disabled.

You can enable this feature if your switch is running PVST+, Rapid PVST+, or MSTP.

By default, PortFast is disabled on all interfaces.

Example: Enabling PortFast

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

spanning-tree portfast [**trunk**]
Enables PortFast on an access port connected to a single workstation or server. By specifying the **trunk** keyword, you can enable PortFast on a trunk port.
To enable PortFast on trunk ports, you must use the **spanning-tree portfast trunk** interface configuration command. The **spanning-tree portfast** command will not work on trunk ports.

You can use the **spanning-tree portfast default** global configuration command to globally enable the PortFast feature on all non-trunking ports.

BPDU Guard

BPDU guard feature can be globally enabled on the switch or can be enabled per port, but the feature operates with some differences.

When you enable BPDU guard at the global level on PortFast enabled ports, spanning tree shuts down ports that are in a PortFast operational state if any BPDU is received on them. In a valid configuration, PortFast enabled ports do not receive BPDUs. Receiving a BPDU on a PortFast enabled port means an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the error-disabled state. When this happens, the switch shuts down the entire port on which the violation occurred.

When you enable BPDU guard at the interface level on any port without also enabling the PortFast edge feature, and the port receives a BPDU, it is put in the error-disabled state.

By default, BPDU guard is disabled.

Example: Enabling BPDU Guard

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

spanning-tree portfast edge bpduguard default
Globally enables BPDU guard.

```
Switch(config)# interface gigabitethernet1/0/2
Switch(config-if)# spanning-tree portfast edge
```

spanning-tree portfast edge
Enables the PortFast edge feature.

BPDU Filter

The BPDU filtering feature can be globally enabled on the switch or can be enabled per interface, but the feature operates with some differences.

Enabling BPDU filtering on PortFast edge-enabled interfaces at the global level keeps those interfaces that are in a PortFast edge-operational state from sending or receiving BPDUs. The interfaces still send a few BPDUs at link-up before the switch begins to filter outbound BPDUs. You should globally enable BPDU filtering on a switch so that hosts connected to these interfaces do not receive BPDUs. If a BPDU is received on a PortFast edge-enabled interface, the interface loses its PortFast edge-operational status, and BPDU filtering is disabled.

Enabling BPDU filtering on an interface without also enabling the PortFast edge feature keeps the interface from sending or receiving BPDUs.

WARNING: Enabling BPDU filtering on an interface is the same as disabling spanning tree on it and can result in spanning-tree loops.

You can enable the BPDU filtering feature for the entire switch or for an interface.

By default, BPDU filtering is disabled.

Example: Enabling BPDU Filtering

```
Switch(config)# spanning-tree portfast edge bpdufilter default
```

spanning-tree portfast edge bpdufilter default
Globally enables BPDU filtering.

```
Switch(config)# interface gigabitethernet1/0/2
Switch(config-if)# spanning-tree portfast edge
```

spanning-tree portfast edge
Enables the PortFast edge feature on the specified interface.

You can also use the **spanning-tree bpdufilter enable** interface configuration command to enable BPDU filtering on any interface without also enabling the PortFast edge feature. This command prevents the interface from sending or receiving BPDUs.

Loop Guard, Root Guard

Reference: Configuring Optional Spanning-Tree Features

Loop Guard

You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is enabled on the entire switched network. Loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

When the switch is operating in PVST+ or rapid-PVST+ mode, loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

When the switch is operating in MST mode, BPDUs are not sent on non-boundary ports only if the interface is blocked by loop guard in all MST instances. On a boundary port, loop guard blocks the interface in all MST instances.

Loop guard operates only on interfaces that are considered point-to-point by the spanning tree.

By default, loop guard is disabled.

NOTE: You cannot enable both loop guard and root guard at the same time.

Example: Enabling Loop Guard

```
Switch(config)# spanning-tree loopguard default
```

spanning-tree loopguard default
Enables loop guard.

Root Guard

The Layer 2 network of a service provider (SP) can include many connections to switches that are not owned by the SP. In such a topology, the spanning tree can reconfigure itself and select a customer switch as the root switch. You can avoid this situation by enabling root guard on SP switch interfaces that connect to switches in your customer's network. If spanning-tree calculations cause an interface in the customer network to be selected as the root port, root guard then places the interface in the root-inconsistent (blocked) state to prevent the customer's switch from becoming the root switch or being in the path to the root.

If a switch outside the SP network becomes the root switch, the interface is blocked (root-inconsistent state), and spanning tree selects a new root switch. The customer's switch does not become the root switch and is not in the path to the root.

Root guard enabled on an interface applies to all the VLANs to which the interface belongs. VLANs can be grouped and mapped to an MST instance.

Root guard enabled on an interface applies to all the VLANs to which the interface belongs. Do not enable the root guard on interfaces to be used by the UplinkFast feature. With UplinkFast, the backup interfaces (in the blocked state) replace the root port in the case of a failure. However, if root guard is also enabled, all the backup interfaces used by the UplinkFast feature are placed in the root-inconsistent state (blocked) and are prevented from reaching the forwarding state.

WARNING: Misuse of the root guard feature can cause a loss of connectivity.

NOTE: You cannot enable both root guard and loop guard at the same time.

By default, root guard is disabled on all interfaces.

Example: Enabling Root Guard

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

spanning-tree guard root
Enables root guard on the interface.