# Configuring Ethernet Switch Ports

## Configuring VLANs

A VLAN is a switched network that is logically segmented by function or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs. However, you can group end-stations even if they are not physically located on the same LAN segment. Any device port can belong to a VLAN, unicast, broadcast, and multicast packets are forwarded and flooded only to end-stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router or a device supporting fallback bridging. In a device stack, VLANs can be formed with ports across the stack. Because a VLAN is considered a separate logical network, it contains its own bridge Management Information Base (MIB) information and can support its own implementation of spanning tree.

VLANs are often associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. Interface VLAN membership on the device is assigned manually on an interface-by-interface basis. When you assign device interfaces to VLANs by using this method, it is known as interface-based, or static, VLAN membership.

The device can route traffic between VLANs by using device virtual interfaces (SVIs). An SVI must be explicitly configured and assigned an IP address to route traffic between VLANs.

### Access Ports

An access port belongs to and carries the traffic of only one VLAN (unless it is configured as a voice VLAN port). Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged packet IEEE 802.1Q tagged), the packet is dropped, and the source address is not learned.

### Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. These trunk port types are supported:

- An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. An IEEE 802.1Q trunk port is assigned a default port VLAN ID (PVID), and all untagged traffic travels on the port default PVID. All untagged traffic and tagged traffic with a NULL VLAN ID are assumed to belong to the port default PVID. A packet with a VLAN ID equal to the outgoing port default PVID is sent untagged. All other traffic is sent with a VLAN tag.

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can become a member of a VLAN only if VTP knows of the VLAN and if the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

For more information on VLANs, see VLAN Configuration Guide, Cisco IOS XE Gibraltar 16.10.x.

# Creating a VLAN

With VTP version 1 and 2, if the device is in VTP transparent mode, you can assign VLAN IDs greater than 1006, but they are not added to the VLAN database.

To configure the Vlan, perform these steps. You can configure the Vlan in access or trunk mode. The procedure is same for the both the modes.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **vlan** *vlan-id*<br>**Example:**<br>`Router(config)# vlan 20` | Enters a VLAN ID, and enters VLAN configuration mode. Enter a new VLAN ID to create a VLAN, or enter an existing VLAN ID to modify that VLAN. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** The available VLAN ID range for this command is 1 to 4094. |
| **Step 3** | **name** *vlan-name*<br>**Example:**<br>`Router(config-vlan)# name test20` | (Optional) Enters a name for the VLAN. If no name is entered for the VLAN, the default is to append the *vlan-id* value with leading zeros to the word VLAN. For example, VLAN0004 is a default VLAN name for VLAN 4. |
| **Step 4** | **exit**<br>**Example:**<br>`Router(config-vlan)# exit` | Returns to configuration mode. |
| **Step 5** | **interface** *interface-id*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Specifies the physical port to be configured, and enter interface configuration mode. |
| **Step 6** | **switchport mode access**<br>**Example:**<br>`Router(config-if)# switchport mode access` | Configures the interface as a VLAN access port. |
| **Step 7** | **switchport access vlan** *vlan id*<br>**Example:**<br>`Router(config-if)# switchport access vlan 20` | Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic. |
| **Step 8** | **end**<br>**Example:**<br>`Router(config-if)# end` | Returns to configuration mode. |

# Configuring LAN Ports for Layer 2 Switching

This section describes how configure all three types of ethernet LAN ports for Layer 2 switching on the Cisco IR8340 routers. The configuration tasks in this section apply to LAN ports on the router.

# Layer 2 LAN Port Modes

The following table lists the Layer 2 LAN port modes and describes how they function on LAN ports.

*Table 1: Layer 2 LAN Port Modes*

| Mode | Function |
|---|---|
| **switchport mode access** | Puts the LAN port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The LAN port becomes a nontrunk port even if the neighboring LAN port does not agree to the change. |
| **switchport mode dynamic desirable** | Makes the LAN port actively attempt to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk** , **desirable** , or **auto** mode. This is the default mode for all LAN ports. |
| **switchport mode dynamic auto** | Makes the LAN port willing to convert the link to a trunk link. The LAN port becomes a trunk port if the neighboring LAN port is set to **trunk** or **desirable** mode. |
| **switchport mode trunk** | Puts the LAN port into permanent trunking mode and negotiates to convert the link into a trunk link. The LAN port becomes a trunk port even if the neighboring port does not agree to the change. |
| **switchport nonegotiate** | Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames. You must configure the neighboring port manually as a trunk port to establish a trunk link. |

**Note** DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that LAN ports connected to devices that do not support DTP are configured with the **access** keyword if you do not intend to trunk across those links. To enable trunking to a device that does not support DTP, use the **nonegotiate** keyword to cause the LAN port to become a trunk but not generate DTP frames.

# Default Layer 2 LAN Interface Configuration

The following table shows the Layer 2 LAN port default configuration.

*Table 2: Layer 2 LAN Interface Default Configuration*

| Feature | Default |
|---|---|
| Interface mode: | |
| • Before entering the **switchport** command | |
| • After entering the **switchport** command | **switchport mode dynamic desirable** |
| Default access VLAN | VLAN 1 |

| Feature | Default |
|---|---|
| Native VLAN (for 802.1Q trunks) | VLAN 1 |

## Configuring LAN Interfaces for Layer 2 Switching

These sections describe how to configure Layer 2 switching on the Cisco IR8340 routers:

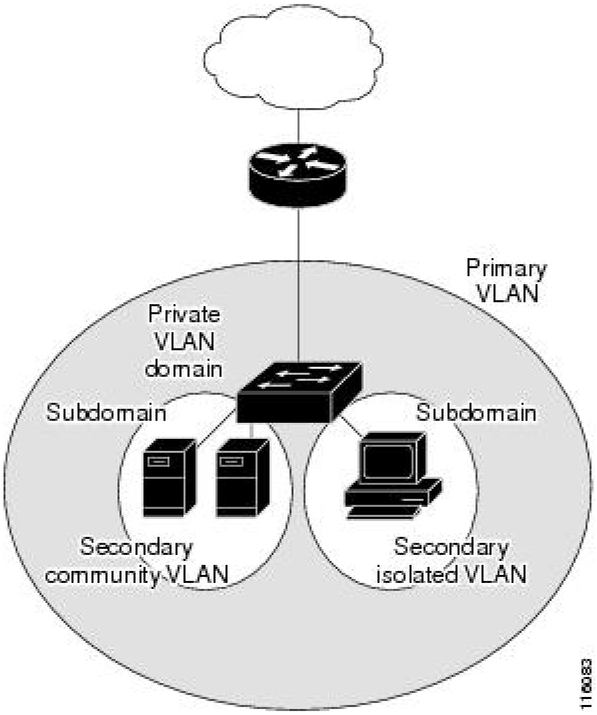
**Note** Use the default **default interface** *interface -type slot/subslot/port* command to revert an interface to its default configuration.

# Configuring Private VLANs

## Information About Private VLANs

Using private VLANs addresses the scalability problem and provides IP address management benefits for service providers and Layer 2 security for customers. Private VLANs partition a regular VLAN domain into subdomains. A subdomain is represented by a pair of VLANs: a primary VLAN and a secondary VLAN. A private VLAN can have multiple VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

***Figure 1: Private VLAN Domain***



## Primary and Secondary VLANs in Private VLANs

A private VLAN domain has only one primary VLAN. Each port in a private VLAN domain is a member of the primary VLAN; the primary VLAN is the entire private VLAN domain.

Secondary VLANs provide isolation between ports within the same private VLAN domain. The following two types are secondary VLANs within a primary VLAN:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate directly with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other community VLANs or in any isolated VLANs at the Layer 2 level.

## Private VLAN Ports

The three types of PVLAN ports are as follows:

- Promiscuous port—A promiscuous port belongs to the primary VLAN. The promiscuous port can communicate with all interfaces, including the community and isolated host ports, that belong to those secondary VLANs associated to the promiscuous port and associated with the primary VLAN. You can have several promiscuous ports in a primary VLAN. Each promiscuous port can have several secondary VLANs or no secondary VLANs that are associated to that port. You can associate a secondary VLAN to more than one promiscuous port, as long as the promiscuous port and secondary VLANs are within the same primary VLAN. You may want to do this for load-balancing or redundancy purposes. You can also have secondary VLANs that are not associated to any promiscuous port.

  A promiscuous port can be configured as an access port.

- Isolated port—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same PVLAN domain, except that it can communicate with associated promiscuous ports. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

  An isolated port can be configured an access port.

- Community port—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with associated promiscuous ports. These interfaces are isolated from all other interfaces in other communities and from all isolated ports within the PVLAN domain.

  A community port must be configured as an access port.

## Guidelines and Limitations for Private VLANs

The following are guidelines and limitations for configuring Private VLANs on IR8340:

- A primary VLAN can have only one isolated VLAN and multiple community VLANs.
- Secondary VLANs must be part of one and only one primary VLAN.
- SVI for secondary VLAN is disabled when primary-secondary VLAN association is configured. Deleting the association will bring back the secondary VLAN SVI.
- Deletion of a secondary VLAN puts the ports in that VLAN in inactive state.

- VLANs that cannot be configured as PVLAN—1, 1001-1005
- All switches in the network must be manually configured with the primary-secondary VLAN association. Otherwise, the MAC addresses will not be replicated from primary VLAN to secondary and vice versa, in that switch. That will lead to flooding of PVLAN traffic.
- Maximum number of primary or secondary VLANs that can be configured is limited by the number of VLANs that can be supported by the switch.
- Maximum number of end devices that can be configured in PVLAN is limited by the L2 TCAM entry limitation.

The following features are supported on IR8340:

- **Isolated access port**—Access port which can only communicate with promiscuous port
- **Promiscuous access port**—Access ports which can communicate with all ports in private VLAN
- **Community access port**—Access ports which can communicate with ports in same community and promiscuous ports
- **Private VLAN across switches**—Private VLAN traffic can be carried across normal trunk ports and the feature can span across switches
- **Multicast in Private VLAN**—Multicast communication in and out of private VLAN

The following features are not supported on IR8340:

- **2-way community VLAN**—The community ports send and receive traffic in the same VLAN
- **Promiscuous trunk port**—A trunk port carrying primary VLAN traffic for multiple private VLAN. The secondary VLANs are explicitly mapped to primary VLAN for multiple private VLAN
- **Trunk isolated/community ports**—Isolated and community ports are trunk with secondary VLANs of multiple private VLAN
- **Layer 3 communication between isolated ports**—Isolated ports can communicate at layer 3

# Configuring a Private VLAN

## Configuring a VLAN as a Private VLAN

To create a private VLAN, you first create a VLAN, and then configure that VLAN to be a private VLAN.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **vlan** *vlan-id*<br>**Example:** | Enters VLAN configuration submode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config)# vlan 202` | |
| **Step 3** | [**no**] **private-vlan** {**community** \| **isolated** \| **primary**}<br>**Example:**<br>`Router(config-vlan)# private-vlan primary` | Configures a VLAN as a private VLAN. Use the no form of this command to clear the private VLAN configuration.<br>**Note** These commands will not take effect until you exit VLAN configuration submode. |
| **Step 4** | **end**<br>**Example:**<br>`Router(config-vlan)# end` | Exits VLAN configuration mode. |
| **Step 5** | **show vlan private-vlan** [*type*]<br>**Example:**<br>`Router# show vlan private-vlan` | Verifies the configuration.<br>**Note** When a VLAN is configured as part of a private VLAN, all the normal access ports belonging to the VLAN will be brought down, waiting for the configuration of their port roles in the PVLAN to take effect. |

### Example

The following example shows how to configure VLAN 202 as a primary VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan primary
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
```

The following example shows how to configure VLAN 303 as a community VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 303
Router(config-vlan)# private-vlan community
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
```

The following example shows how to configure VLAN 440 as an isolated VLAN and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 440
Router(config-vlan)# private-vlan isolated
Router(config-vlan)# end
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202               primary
        303       community
        440       isolated
```

## Associating Secondary VLANs with a Primary Private VLAN

To associate secondary VLANs with a primary VLAN, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **vlan** *primary-vlan-id*<br>**Example:**<br>`Router(config)# vlan 202` | Enters VLAN configuration submode for the primary VLAN. |
| **Step 3** | **private-vlan association** {*secondary_vlan_list* \| **add** *secondary_vlan_list* \| **remove** *secondary_vlan_list*}<br>**Example:**<br>`Router(config-vlan)# private-vlan association 303-307,309,440` | Associates the secondary VLANs with the primary VLAN.<br>Use **no private-vlan association** to clear all secondary VLAN associations. |
| **Step 4** | **end**<br>**Example:**<br>`Router(config-vlan)# end` | Exits VLAN configuration mode. |
| **Step 5** | **show vlan private-vlan** [*type*]<br>**Example:**<br>`Router# show vlan private-vlan` | Verifies the configuration. |

### Example

The following example shows how to associate community VLANs 303 through 307, 309, and isolated VLAN 440 with primary VLAN 202, and verify the configuration:

```
Router# configure terminal
Router(config)# vlan 202
Router(config-vlan)# private-vlan association 303-307,309,440
Router(config-vlan)# end
```

```
Router# show vlan private-vlan
Primary Secondary Type              Interfaces
------- --------- ----------------- ------------------------------------------
202     303       community
202     304       community
202     305       community
202     306       community
202     307       community
202     309       community
202     440       isolated
        308       community
```

## Configuring a Layer 2 Interface as a Private VLAN Host Port

To configure a Layer 2 interface as a private VLAN host port, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Selects the LAN interface to configure. |
| **Step 3** | **switchport**<br>**Example:**<br>`Router(config-if)# switchport` | Configures the LAN interface for Layer 2 switching. |
| **Step 4** | **switchport mode private-vlan host**<br>**Example:**<br>`Router(config-if)# switchport mode private-vlan host` | Configures the Layer 2 port as a private VLAN host port.<br>Use **no switchport mode private-vlan host** to clears private VLAN port configuration. |
| **Step 5** | **switchport private-vlan host host-association** *primary_vlan_ID secondary_vlan_ID*<br>**Example:**<br>`Router(config-if)# switchport private-vlan host-association 202 303` | Associates the Layer 2 port with a private VLAN.<br>Use **no switchport private-vlan host host-association** to clear the association. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config-if)# end` | Exits interface configuration mode. |
| **Step 7** | **show interface** *type slot/port* **switchport**<br>**Example:** | Verifies the configuration. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router# show interface gigabitethernet`<br>`0/1/0 switchport` | |

### Example

This example shows how to configure interface gigabitethernet 0/1/0 as a private VLAN host port and verify the configuration:

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# switchport mode private-vlan host
Router(config-if)# switchport private-vlan host-association 202 303
Router(config-if)# end
Router# show interfaces gigabitethernet 0/1/0 switchport
Name: Ge0/1/0
Switchport: Enabled
 Administrative Mode: private-vlan host
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
 Administrative private-vlan host-association: 202 (VLAN0202) 303 (VLAN0303)
Administrative private-vlan mapping: none
 Operational private-vlan: 202 (VLAN0202)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

## Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port

To configure a Layer 2 interface as a private VLAN promiscuous port, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *type slot/port*<br>**Example:**<br>`Router(config)# interface gigabitethernet`<br>`0/1/0` | Selects the LAN interface to configure. |
| **Step 3** | **switchport**<br>**Example:**<br>`Router(config-if)# switchport` | Configures the LAN interface for Layer 2 switching. |
| **Step 4** | **switchport mode private-vlan promiscuous**<br>**Example:** | Configures the Layer 2 port as a private VLAN promiscuous port. |

| | Command or Action | Purpose |
|---|---|---|
| | Router(config-if)# **switchport mode private-vlan promiscuous** | Use **no switchport mode private-vlan** to clear the private VLAN port configuration. |
| **Step 5** | **switchport private-vlan mapping** *primary_vlan_ID* {*secondary_vlan_list* \| **add** *secondary_vlan_list* \| **remove** *secondary_vlan_list*} *secondary_vlan_ID*<br>**Example:**<br>Router(config-if)# **switchport private-vlan mapping 202 303,440** | Maps the private VLAN promiscuous port to a primary VLAN and to selected secondary VLANs.<br>Use **no switchport private-vlan mapping** to clear all mapping between the private VLAN promiscuous port and the primary VLAN and any secondary VLANs. |
| **Step 6** | **end**<br>**Example:**<br>Router(config-if)# **end** | Exits interface configuration mode. |
| **Step 7** | **show interface** *type slot/port* **switchport**<br>**Example:**<br>Router# **show interface gigabitethernet 0/1/0 switchport** | Verifies the configuration. |

### Example

The following example shows how to configure interface gigabitethernet 0/1/0 as a private VLAN promiscuous port and map it to a private VLAN:

```
Router# configure terminal
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# switchport mode private-vlan promiscuous
Router(config-if)# switchport private-vlan mapping 202 303,440
Router(config-if)# end
```

The following example shows how to verify the configuration:

```
Router# show interfaces gigabitethernet 0/1/0 switchport
Name: Ge0/1/0
Switchport: Enabled
 Administrative Mode: private-vlan promiscuous
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative private-vlan host-association: none ((Inactive))
 Administrative private-vlan mapping: 202 (VLAN0202) 303 (VLAN0303) 440 (VLAN0440)
 Operational private-vlan: 202 (VLAN0202)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
```

# Configuring Voice VLANs

The voice VLAN feature provides support for connecting an IP phone to an access switch port. Voice VLAN on an access port is desirable so that feature like port security, dot1x, dynamic access port, protected port can be configured.

The voice VLAN requires the access port to support dedicated VLAN for voice traffic (as the data traffic on the phone link might deteriorate the voice traffic quality) thus the device can differentiate voice traffic from data traffic and provide QoS for voice traffic and ensure quality.

The Ethernet port will be associated with two VLANs on a voice VLAN port as following:

1. A native VLAN to carry data traffic
2. An auxiliary or Voice VLAN to carry voice traffic

The data traffic will be sent either tagged or untagged with the access VLAN id. The phone will send voice traffic tagged with configured voice VLAN id. The voice VLAN id used by phone can either configured manually or learned through CDP. When voice VLAN is configured on the access port, the device will instruct IP phone to send voice traffic over the configured voice VLAN. This is achieved through sending CDP messages to IP phone indicating the same. QoS configurations can be done on the voice VLAN port in order to provide predictable forwarding of voice traffic and thus ensure voice quality.

## Limitations and Restrictions

- Voice VLAN configuration is only supported on device access ports.
- Voice VLAN configuration will not be applicable to port channels.
- Private VLAN configuration will not be allowed on a voice VLAN port and vice versa.

## How to Configure Voice VLANs

The following sections provide information about configuring Voice VLANs:

### Configuring Cisco IP Phone Voice Traffic

You can configure a port connected to the Cisco IP Phone to send CDP packets to the phone to configure the way in which the phone sends voice traffic. The phone can carry voice traffic in IEEE 802.1Q frames for a specified voice VLAN with a Layer 2 CoS value. It can use IEEE 802.1p priority tagging to give voice traffic a higher priority and forward all voice traffic through the native (access) VLAN. The Cisco IP Phone can also send untagged voice traffic or use its own configuration to send voice traffic in the access VLAN. In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5).

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:** | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router#` **`configure terminal`** | |
| **Step 2** | **interface** *interface-id*<br>**Example:**<br>`Router(config)#` **`interface gigabitethernet 0/1/0`** | Specifies the interface connected to the phone, and enters interface configuration mode. |
| **Step 3** | **switchport voice vlan** {*vlan-id* \| **dot1p** \| **none** \| **untagged**}<br>**Example:**<br>`Router(config-if)#` **`switchport voice vlan dot1p`** | Configures the voice VLAN.<br>• *vlan-id* —Configures the phone to forward all voice traffic through the specified VLAN. By default, the Cisco IP Phone forwards the voice traffic with an IEEE 802.1Q priority of 5. Valid VLAN IDs are 1 to 4094.<br>• **dot1p** —Configures the phone to use IEEE 802.1p priority tagging for voice traffic and to use the default native VLAN (VLAN 0) to carry all traffic. By default, the Cisco IP Phone forwards the voice traffic with an IEEE 802.1p priority of 5.<br>• **none** —Allows the phone to use its own configuration to send untagged voice traffic.<br>• **untagged** —Configures the phone to send untagged voice traffic. |
| **Step 4** | **end**<br>**Example:**<br>`Router(config-if)#` **`end`** | Exits interface configuration mode. |

### What to do next

To configure port security for Voice VLAN:

```
Router#configure terminal
Router(config)#interface <interface-id>
Router(config-if)# switchport mode access
Router(config-if)# switchport voice vlan vlan-id
Router(config-if)# switchport port-security
Router(config-if)# switchport port-security [maximum value [vlan {access | voice}}]]
Router(config-if)# switchport port-security violation {protect | restrict | shutdown}
Router(config-if)# switchport port-security [mac-address mac-address [vlan {access | voice}]]
Router(config-if)#end
Router#show port-security
```

## Removing Voice VLAN

To remove the voice VLAN configuration, use the **no switchport voice vlan** command.

## Monitoring Voice VLAN

To display voice VLAN configuration for an interface, use the **show interfaces** *interface-id* **switchport** privileged EXEC command.

# Configuring VXLAN Tunneling

VXLAN is an extension to the Layer 2 VLAN. It was designed to provide the same VLAN functionality with greater extensibility and flexibility. VXLAN offers the following benefits:

- VLAN flexibility in multitenant segments: It provides a solution to extend Layer 2 segments over the underlying network infrastructure so that tenant workload can be placed across physical pods in the data center.
- Higher scalability: VXLAN uses a 24-bit segment ID known as the VXLAN network identifier (VNID), which enables up to 16 million VXLAN segments to coexist in the same administrative domain.
- Improved network utilization: VXLAN solved Layer 2 STP limitations. VXLAN packets are transferred through the underlying network based on its Layer 3 header and can take complete advantage of Layer 3 routing, equal-cost multipath (ECMP) routing, and link aggregation protocols to use all available paths.

VXLAN uses the VXLAN tunnel endpoint (VTEP) to map tenants' end devices to VXLAN segments and to perform VXLAN encapsulation and decapsulation. Each VTEP function has two interfaces: one is a switch interface on the local LAN segment to support local endpoint communication, and the other is an IP interface to the transport IP network.

Infrastructure VLAN is a unique IP address that identifies the VTEP device on the transport IP network. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface.

A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface.

The following example shows that two IR8340 routers act as VTEPs:

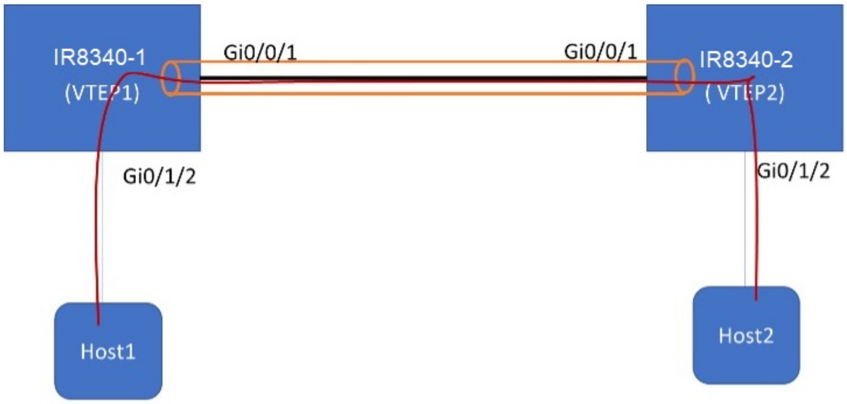


*Table 3: VXLAN Configuration*

| IR8340-1 | IR8340-2 |
|---|---|
| `bridge-domain 1`<br>`member vni 6001`<br>`member Vlan100 service-instance 1`<br>`!`<br>`interface Loopback1`<br>`ip address 200.200.200.200 255.255.255.255`<br>`!`<br>`interface GigabitEthernet0/0/1`<br>`ip address 192.168.1.2 255.255.255.0`<br>`media-type rj45`<br>`!`<br>`Interface GigabitEthernet0/1/2`<br>`switchport access vlan 100`<br>`!`<br>`interface Vlan100`<br>`no ip address`<br>`service instance 1 ethernet`<br>`  encapsulation dot1q 100`<br>`!`<br>`interface nve1`<br>`no ip address`<br>`source-interface Loopback1`<br>`member vni 6001`<br>`  ingress-replication 100.100.100.100`<br>`!`<br>`ip forward-protocol nd`<br>`ip pim rp-address 200.200.200.200`<br>`ip http server`<br>`ip http secure-server`<br>`ip route 0.0.0.0 0.0.0.0 192.168.1.3`<br>`!` | `bridge-domain 1`<br>`member vni 6001`<br>`member Vlan100 service-instance 1`<br>`!`<br>`interface Loopback1`<br>`ip address 100.100.100.100 255.255.255.255`<br>`!`<br>`interface GigabitEthernet0/0/1`<br>`ip address 192.168.1.3 255.255.255.0`<br>`media-type rj45`<br>`!`<br>`interface GigabitEthernet0/1/2`<br>`switchport access vlan 100`<br>`!`<br>`interface Vlan100`<br>`no ip address`<br>`service instance 1 ethernet`<br>`encapsulation dot1q 100`<br>`!`<br>`interface nve1`<br>`no ip address`<br>`source-interface Loopback1`<br>`member vni 6001`<br>`  ingress-replication 200.200.200.200`<br>`!`<br>`ip forward-protocol nd`<br>`ip pim rp-address 100.100.100.100`<br>`no ip http server`<br>`ip http secure-server`<br>`ip route 0.0.0.0 0.0.0.0 192.168.1.2`<br>`!` |

For more details and multiple VTEP configuration with multicast, see https://www.cisco.com/c/en/us/support/docs/ip/multicast/200791-Configuration-and-Troubleshooting-of-VxL.html.

# IEEE 802.1x Protocol

The IEEE 802.1x standard defines a client/server-based access control and authentication protocol that prevents clients from connecting to a LAN through publicly accessible ports unless they are authenticated. The authentication server authenticates each client connected to a port before making available any services offered by the router or the LAN.

Until the client is authenticated, IEEE 802.1x access control allows only Extensible Authentication Protocol over LAN (EAPOL), Cisco Discovery Protocol (CDP), and Spanning Tree Protocol (STP) traffic through the port to which the client is connected. After authentication, normal traffic can pass through the port. For more information on IEEE 802.1x port-based authentication, see the Configuring IEEE 802.1x Port-Based Authentication chapter of the *Security Configuration Guide, Cisco IOS XE Gibraltar 16.10.x*.

## Configuring IEEE 802.1X Port-Based Authentication

IEEE 802.1X port-based authentication is configured on a device to prevent unauthorized devices (supplicants) from gaining access to the network. The device can combine the function of a router, switch, and access point, depending on the fixed configuration or installed modules. The switch functions are provided by either built-in switch ports or a plug-in module with switch ports. This feature supports both access ports and trunk ports. For more informaton on 802.1X port-based authentication, see the Configuring IEEE 802.1X Port-Based Authentication Guide.

# Enabling AAA Authorization for VLAN Assignment

AAA authorization limits the services available to a user. When AAA authorization is enabled, the device uses information retrieved from the user's profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **aaa new-model**<br>**Example:**<br>`Router(config)# aaa new-model` | Enables AAA. |
| **Step 4** | **aaa authorization network radius if-authenticated**<br>**Example:**<br>`Router(config)# aaa authorization network radius if-authenticated` | Configures the device for user RADIUS authorization for all network-related service requests. RADIUS authorization succeeds if the user has authenticated. |
| **Step 5** | **aaa authorization exec radius if-authenticated**<br>**Example:**<br>`Router(config)# aaa authorization exec radius if-authenticated` | Configures the device for user RADIUS authorization if the user has privileged EXEC access. RADIUS authorization succeeds if the user has authenticated. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

# Enabling IEEE 802.1X Authentication and Authorization

Follow these steps to enable IEEE 802.1X authentication and authorization.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **aaa authentication dot1x** {**default** \| **listname**} **method1** [**method2**...]<br>**Example:**<br>`Router(config)# aaa authentication dot1x default group radius` | Creates a series of authentication methods that are used to determine user privilege to access the privileged command level so that the device can communicate with the AAA server. |
| **Step 4** | **dot1x system-auth-control**<br>**Example:**<br>`Router(config)# dot1x system-auth-control` | Globally enables 802.1X port-based authentication. |
| **Step 5** | **identity profile default**<br>**Example:**<br>`Router(config)# identity profile default` | Creates an identity profile and enters dot1x profile configuration mode. |
| **Step 6** | **exit**<br>**Example:**<br>`Router(config-identity-prof)# exit` | Exits dot1x profile configuration mode and returns to global configuration mode. |
| **Step 7** | **interface** *type slot/port*<br>**Example:**<br>`Router(config)# interface Gigabitethernet 0/1/0` | Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication. |
| **Step 8** | **access-session port-control** {**auto** \| **force-authorized** \| **force-unauthorized**}<br>**Example:**<br>`Router(config-if)# access-session port-control auto` | Enables 802.1X port-based authentication on the interface.<br>• **auto** —Enables IEEE 802.1X authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port changes from down to up or when an EAPOL-start frame is received. The Device requests the identity of the supplicant and begins relaying authentication messages between the |

| | Command or Action | Purpose |
|---|---|---|
| | | supplicant and the authentication server. Each supplicant attempting to access the network is uniquely identified by the Device by using the supplicant MAC address.<br>• **force-authorized** -—Disables IEEE 802.1X authentication and causes the port to change to the authorized state without any authentication exchange required. The port sends and receives normal traffic without IEEE 802.1X-based authentication of the client. This is the default setting.<br>• **force-unauthorized** —Causes the port to remain in the unauthorized state, ignoring all attempts by the supplicant to authenticate. The Device cannot provide authentication services to the supplicant through the port. |
| **Step 9** | **dot1x pae** [**supplicant** \| **authenticator** \| **both**]<br>**Example:**<br>`Device(config-if)# dot1x pae authenticator` | Sets the Port Access Entity (PAE) type.<br>• **supplicant** —The interface acts only as a supplicant and does not respond to messages that are meant for an authenticator.<br>• **authenticator** -—The interface acts only as an authenticator and does not respond to any messages meant for a supplicant.<br>• **both** —The interface behaves both as a supplicant and as an authenticator and thus does respond to all dot1x messages. |
| **Step 10** | **end**<br>**Example:**<br>`router(config)# end` | Returns to privileged EXEC mode. |
| **Step 11** | **show dot1x**<br>**Example:**<br>`Router# show dot1x` | Displays whether 802.1X authentication has been configured on the device. |

# Spanning Tree Protocol Overview

Spanning Tree Protocol (STP) is a Layer 2 link management protocol that provides path redundancy while preventing loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. Multiple active paths among end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages. Device might also learn end-station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network. Spanning-tree operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

The STP uses a spanning-tree algorithm to select one device of a redundantly connected network as the root of the spanning tree. The algorithm calculates the best loop-free path through a switched Layer 2 network by assigning a role to each port based on the role of the port in the active topology:

- Root—A forwarding port elected for the spanning-tree topology
- Designated—A forwarding port elected for every switched LAN segment
- Alternate—A blocked port providing an alternate path to the root bridge in the spanning tree
- Backup—A blocked port in a loopback configuration

The device that has *all* of its ports as the designated role or as the backup role is the root device. The device that has at least *one* of its ports in the designated role is called the designated device.

Spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path. Device send and receive spanning-tree frames, called bridge protocol data units (BPDUs), at regular intervals. The device do not forward these frames but use them to construct a loop-free path. BPDUs contain information about the sending device and its ports, including device and MAC addresses, device priority, port priority, and path cost. Spanning tree uses this information to elect the root device and root port for the switched network and the root port and designated port for each switched segment.

When two ports on a device are part of a loop, the spanning-tree and path cost settings control which port is put in the forwarding state and which is put in the blocking state. The spanning-tree port priority value represents the location of a port in the network topology and how well it is located to pass traffic. The path cost value represents the media speed.

**Note** By default, the device sends keepalive messages (to ensure the connection is up) only on interfaces that do not have small form-factor pluggable (SFP) modules. You can change the default for an interface by entering the [**no**] **keepalive** interface configuration command with no keywords.

IR8340 uses STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

For more information on STP, see the **Configuring Spanning Tree Protocol** chapter of the Layer 2 Configuration Guide.

# Default STP Configuration

The following table shows the default STP configuration.

*Table 4: STP Default Configuration*

| Feature | Default Value |
|---|---|
| Disable state | STP disabled for all VLANs |
| Bridge priority | 32768 |
| STP port priority (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports) | 128 |
| STP port cost (configurable on a per-port basis—used on LAN ports configured as Layer 2 access ports) | Gigabit Ethernet: 4 |
| STP VLAN port priority (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports) | 128 |
| STP VLAN port cost (configurable on a per-VLAN basis—used on LAN ports configured as Layer 2 trunk ports) | Gigabit Ethernet:1000000000 |
| Hello time | 2 seconds |
| Forward delay time | 15 seconds |
| Maximum aging time | 20 seconds |
| Mode | PVST |

# Enabling STP

**Note** STP is disabled by default on all VLANs.

You can enable STP on a per-VLAN basis. The Cisco SM-X-16G4M2X or SM-X-40G8M2X Layer 2 Gigabit EtherSwitch Service Module maintain a separate instance of STP for each VLAN (except on VLANs on which you disable STP).

If you want to enable a mode that is different from the default mode, this procedure is required.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **spanning-tree mode** {**pvst** \| **mst** \| **rapid-pvst**} | Configures a spanning-tree mode.<br>All stack members run the same version of spanning tree.<br>• Select **pvst** to enable PVST+.<br>• Select **mst** to enable MSTP.<br>• Select **rapid-pvst** to enable rapid PVST+. |
| **Step 3** | **interface** *interface-id* | Specifies an interface to configure, and enters interface configuration mode. Valid interfaces include physical ports, VLANs, and port channels. The VLAN ID range is 1 to 4094. The port-channel range is 1 to 48. |
| **Step 4** | **spanning-tree link-type point-to-point**<br>**Example:**<br>`Device(config-if)# spanning-tree link-type point-to-point` | Specifies that the link type for this port is point-to-point.<br>If you connect this port (local port) to a remote port through a point-to-point link and the local port becomes a designated port, the negotiates with the remote port and rapidly changes the local port to the forwarding state. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| **Step 6** | **clear spanning-tree detected-protocols**<br>**Example:**<br>`Router# clear spanning-tree detected-protocols` | If any port on the device is connected to a port on a legacy IEEE 802.1D device, this command restarts the protocol migration process on the entire device.<br>This step is optional if the designated device detects that this device is running rapid PVST+. |
| **Step 7** | **show spanning-tree vlan** *vlan_ID* | Verifies that STP is enabled. |

### What to do next

**Caution** Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.

**Caution** We do not recommend disabling spanning tree, even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN.

This example shows how to enable STP on VLAN 200:

```
Device# configure terminal
Device(config)# spanning-tree vlan 200
Device(config)# end
Device#
```

**Note** STP is disabled by default.

This example shows how to verify the configuration:

```
Device# show spanning-tree vlan 200

G0:VLAN0200
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     00d0.00b8.14c8
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32768
             Address     00d0.00b8.14c8
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300
Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
Gi1/4            Desg FWD 200000    128.196  P2p
Gi1/5            Back BLK 200000    128.197  P2p
Device#
```

You must have at least one interface that is active in VLAN 200 to create a VLAN 200 spanning tree. In this example, two interfaces are active in VLAN 200.

# Configuring Optional STP Features

This section describes how to configure the following optional STP features:

## Enabling PortFast

**Caution** Use PortFast *only* when connecting a single end station to a Layer 2 access port. Otherwise, you might create a network loop.

To enable PortFast on a Layer 2 access port, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | Router(config)# **interface** {*type slot/port*} | Selects a port to configure. |
| **Step 2** | Router(config-if)# **spanning-tree portfast** | Enables PortFast on a Layer 2 access port connected to a single workstation or server. |
| **Step 3** | Router(config-if)# **spanning-tree portfast default** | Enables PortFast. |
| **Step 4** | Router(config-if)# **end** | Exits configuration mode. |
| **Step 5** | Router# **show running interface** {*type slot/port*} | Verifies the configuration. |

## Configuring PortFast BPDU Filtering

These sections describe how to configure PortFast BPDU filtering.

To enable PortFast BPDU filtering globally, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **spanning-tree portfast bpdufilter default**<br>**Example:**<br>`Router(config)# spanning-tree portfast bpdufilter default` | Enables BPDU filtering globally on the router. |
| **Step 2** | **show spanning-tree summary totals**<br>**Example:**<br>`Router(config)# show spanning-tree summary totals` | Verifies the configuration. |

### Enabling PortFast BPDU Filtering

BPDU filtering is set to default on each port. This example shows how to enable PortFast BPDU filtering on the port and verify the configuration in PVST+ mode:

```
Router(config)# spanning-tree portfast bpdufilter default

Router(config)# ^Z
Router# show spanning-tree summary totals

Switch is in pvst mode
Root bridge for: G0:VLAN0013, G0:VLAN0020, G1:VLAN0020
EtherChannel misconfig guard is enabled
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
UplinkFast                   is disabled
```

```
BackboneFast                 is disabled
Pathcost method used         is short
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
3 vlans                      0         0        0          3          3
```

To enable PortFast BPDU filtering on a nontrunking port, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **interface** *interface-id*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Selects the interface to configure. |
| **Step 2** | **spanning-tree bpdufilter enable**<br>**Example:**<br>`Router(config-if)# spanning-tree bpdufilter enable` | Enables BPDU filtering. |
| **Step 3** | **show spanning-tree interface** *interface-id*<br>**Example:**<br>`Router# show spanning-tree interface gigabitethernet 0/1/0` | Verifies the configuration. |

### What to do next

This example shows how to enable PortFast BPDU filtering on a nontrunking port:

```
Router(config)# interface gigabitethernet 0/1/0
Router(config-if)# spanning-tree bpdufilter enable

Router(config-if)# ^Z
Router# show spanning-tree interface gigabitethernet 0/1/0
Vlan             Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------------
VLAN0010         Desg FWD 1000      160.196  Edge P2p
Router# show spanning-tree interface gigabitethernet 0/1/0 detail

 Port 196 (gigabitethernet 0/1/0) of VLAN0010 is forwarding
   Port path cost 1000, Port priority 160, Port Identifier 160.196.
   Designated root has priority 32768, address 00d0.00b8.140a
   Designated bridge has priority 32768, address 00d0.00b8.140a
   Designated port id is 160.196, designated path cost 0
   Timers:message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state:1
   The port is in the portfast mode by portfast trunk configuration
   Link type is point-to-point by default
   Bpdu filter is enabled
   BPDU:sent 0, received 0
Router#
```

## Enabling BPDU Guard

To enable BPDU Guard globally, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **spanning-tree portfast bpduguard default**<br>**Example:**<br>`Router(config)# no spanning-tree portfast bpduguard default` | Enables BPDU Guard globally.<br>Disables BPDU Guard globally. |
| **Step 2** | **end**<br>**Example:**<br>`Router(config)# end` | Exits configuration mode. |
| **Step 3** | **show spanning-tree summary totals**<br>**Example:**<br>`Router# show spanning-tree summary totals` | Verifies the configuration. |

### What to do next

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast bpduguard
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals
 default
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast             is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard            is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long
Name                   Blocking Listening Learning Forwarding STP Active
---------------------- -------- --------- -------- ---------- ----------
2 vlans                   0        0         0        3          3
Router#
```

## Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the STP port cost of all Layer 2 LAN interfaces on the device, decreasing the probability that the router will become the root bridge. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second). UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan** *vlan_ID* **priority** command in global configuration mode.

**Note** When you enable UplinkFast, it affects all VLANs on the device. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | Router(config)# **spanning-tree uplinkfast** [**max-update-rate** *max_update_rate*] | Enables UplinkFast. |
| **Step 2** | Router(config)# **no spanning-tree uplinkfast max-update-rate** | Reverts to the default rate. |
| **Step 3** | Router(config)# **no spanning-tree uplinkfast** | Disables UplinkFast. |
| **Step 4** | Router(config)# **end** | Exits configuration mode. |
| **Step 5** | Router# **show spanning-tree vlan** *vlan_ID* | Verifies that UplinkFast is enabled. |

### What to do next

This example shows how to enable UplinkFast with an update rate of 400 packets per second:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast max-update-rate 400
Router(config)# exit
Router#
```

This example shows how to verify that UplinkFast is enabled:

```
Router# show spanning-tree uplinkfast

UplinkFast is enabled
Router#
```

## Enabling BackboneFast

**Note** BackboneFast operates correctly only when enabled on all network devices in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party network devices.

To enable BackboneFast, perform this task:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | Router(config)# **spanning-tree backbonefast** | Enables backbonefast. |
| **Step 2** | Router(config)# **no spanning-tree backbonefast** | Disables BackboneFast. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | Router(config)# **end** | Exits configuration mode. |
| **Step 4** | Router# **show spanning-tree vlan** *vlan_ID* | Verifies that BackboneFast is enabled. |

### What to do next

This example shows how to enable BackboneFast:

```
Router# configure terminal
Router(config)# spanning-tree backboneFast
Router(config)# end
Router#
```

This example shows how to verify that BackboneFast is enabled:

```
Router# show spanning-tree backbonefast

BackboneFast is enabled
BackboneFast statistics
-----------------------
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)     : 0
Number of RLQ request PDUs received (all VLANs)   : 0
Number of RLQ response PDUs received (all VLANs)  : 0
Number of RLQ request PDUs sent (all VLANs)       : 0
Number of RLQ response PDUs sent (all VLANs)      : 0
Router#
```

# MAC Table Manipulation

This section includes the following:

## Creating a Static Entry in the MAC Address Table

Perform the following task to create a static entry in the MAC address table.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **mac address-table static mac-address vlan** *vlan-id* **interface** *Interface-id*<br>**Example:** | Creates a static entry in the MAC address table. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config)# mac address-table static 00ff.ff0d.2dc0 vlan 1 interface gigabitethernet 0/1/0` | |
| **Step 4** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| **Step 5** | **show mac address-table**<br>**Example:**<br>`Router# show mac address-table` | Verifies the MAC address table. |

# MAC Address-Based Traffic Blocking

Perform the following task to block all traffic to or from a MAC address in a specified VLAN.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>- Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **mac address-table static mac-address vlan** *vlan-id* **drop**<br>**Example:**<br>`Router(config)# mac address-table static 00ff.ff0d.2dc0 vlan 1 drop` | Creates a static entry with drop action in the MAC address table. |
| **Step 4** | **end**<br>**Example:**<br>`router(config)# end` | Returns to privileged EXEC mode. |
| **Step 5** | **show mac address-table**<br>**Example:**<br>`Router# show mac address-table` | Verifies the MAC address table. |

# Configuring and Verifying the Aging Timer

Perform this task to configure the aging timer.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **mac address-table aging-time** *time*<br>**Example:**<br>`Router(config)# mac address-table aging-time 600`<br>`Or`<br>`Router(config)# mac address-table aging-time 0` | Configures the MAC address aging timer age in seconds.<br>• The accept value is either 0 or 10-1000000 seconds. Default value is 300 seconds.<br>• The maximum aging timer supported by switch chipset is 634 seconds. If configure greater than 634 seconds, MAC address will age out after 634 seconds.<br>• The value 0 means dynamic MAC entries will never age out. |
| **Step 4** | **end**<br>**Example:**<br>`router(config)# end` | Returns to privileged EXEC mode. |
| **Step 5** | **show mac address-table aging-time**<br>**Example:**<br>`Router# show mac address-table aging-time` | Verifies the MAC address table. |

# MAC Learning on a Vlan

To disable or enable MAC learning on specified vlan, perform these steps.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **mac address-table learning vlan** *vlan-id* **interface** *Interface-id*<br>**Example:**<br>`Router(config)# mac address-table learning vlan 10` | Creates a static entry in the MAC address table. |
| **Step 4** | **end**<br>**Example:**<br>`router(config)# end` | Returns to privileged EXEC mode. |

# Assigning IP Addresses to Switch Virtual Interfaces

To configure IP routing, you need to assign IP addresses to Layer 3 network interfaces. This enables communication with the hosts on those interfaces that use IP. IP routing is disabled by default, and no IP addresses are assigned to Switch Virtual Interfaces (SVIs).

An IP address identifies a destination for IP packets. Some IP addresses are reserved for special uses and cannot be used for host, subnet, or network addresses. RFC 1166, “Internet Numbers,” contains the official description of these IP addresses.

An interface can have one primary IP address. A a subnet mask identifies the bits that denote the network number in an IP address.

Beginning in privileged EXEC mode, follow these steps to assign an IP address and a network mask to an SVI.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal** | Enter global configuration mode. |
| **Step 2** | **interface vlan** *vlan-id* | Enter interface configuration mode, and specify the Layer 3 VLAN to configure. |
| **Step 3** | **ip address** *ip-address subnet-mask* | Configure the IP address and IP subnet mask. |
| **Step 4** | **end** | Returns to privileged EXEC mode. |
| **Step 5** | **show interfaces** [*interface-id*] **show ip interface** [*interface-id*] **show running-config interface** [*interface-id*] | Verify your entries. |
| **Step 6** | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

# SVI Supported Features

The following table provided the supported features on the SVI.

*Table 5: SVI Supported Features*

| Techology | Feature | Use Case |
|---|---|---|
| Routing | Routing Protocol | Interconnects Layer 3 networks using protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF) Protocol, and Enhanced Interior Gateway Routing Protocol (EIGRP) configured under SVI.<br>For more informaton on routing protocol, see the IP Routing: Protocol-Independent Configuration Guide. |
| | Hot Standby Router Protocol (HSRP) | Supports redundancy and high availability with a secondary device connected to the LAN with SVI, using HSRP.<br>For more informaton on HSRP, see the First Hop Redundancy Protocols Configuration Guide. |
| | DHCP | Cisco devices running Cisco software include Dynamic Host Configuration Protocol (DHCP) server and the relay agent software. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the device to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the Domain Name System (DNS) server and the default device.<br>For more informaton on HSRP, see the, IP Addressing: DHCP Configuration Guide |
| | Multicast (IPv4) | Provides multicast support for clients connected to the switch ports.<br>For more informaton on HSRP, see the, IP Multicast: PIM Configuration Guide |
| | VRF | Associates a VRF instance with an SVI to map VLANs to different logical or physical VPN WAN connections.<br>For more informaton on VRF protocol, see the IP Routing: Protocol-Independent Configuration Guide. |

| Techology | Feature | Use Case |
|---|---|---|
| Security | ACL | Provides packet filtering to control network traffic and restrict the access of users and devices to the network<br>For more informaton on ACL protocol, see the Security Configuration Guide: Access Control Lists. |
| | NAT | Provides NAT under SVI.<br>For more information on NAT, see the IP Addressing: NAT Configuration Guide. |
| Qos | Classification with standard and extended access list | Provides QoS classification with standard and extended access lists.<br>For more informtion on QoS, see the Security Configuration Guide: Access Control Lists. |
| | Class-based marking | Provides QoS marking based on user-defined traffic class with DSCP and IP precedence values.<br>For more information on QoS Marking, see the QoS: Classification Configuration Guide. |
| | Policing | Limits the input or output transmission rate on SVI and specifies traffic handling policies when the traffic either conforms to or exceeds the specified rate limits.<br>For more informtion on Policing, see the QoS: Policing and Shaping Configuration Guide |
| Bridging | EVC under SVI | Supports a default encapsulation EFP under SVI, to have VLAN/BD integrated. |
| | EVC with MAC ACL under SVI | For more information on EVC, see *Layer 2 Configuration Guide, Cisco IOS XE Gibraltar 16.11*. |

# IGMP Snooping for IPv4

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content. You can configure the switch to use IGMP snooping in subnets that receive IGMP queries from either IGMP or the IGMP snooping querier. IGMP snooping constrains IPv4 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv4 multicast traffic only to those ports that want to receive it.

Layer 2 switches can use IGMP snooping to constrain the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. As the name implies, IGMP snooping requires the LAN switch to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. When the switch receives an IGMP report from a host for a particular multicast group, the switch adds the host port number to the forwarding table entry; when it receives an IGMP Leave Group message from a host, it removes the host port from the table entry. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients. For more information on this feature, see https://www.cisco.com/c/en/us/td/docs/routers/7600/ios/15S/configuration/guide/7600_15_0s_book/snooigmp.html.

# IGMP Filtering and Throttling

In some environments, for example, metropolitan or multiple-dwelling unit (MDU) installations, you might want to control the set of multicast groups to which a user on a switch port can belong. You can control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan. You might also want to limit the number of multicast groups to which a user on a switch port can belong.

With the IGMP filtering feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual switch ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If an IGMP profile denying access to a multicast group is applied to a switch port, the IGMP join report requesting the stream of IP multicast traffic is dropped, and the port is not allowed to receive IP multicast traffic from that group. If the filtering action permits access to the multicast group, the IGMP report from the port is forwarded for normal processing. You can also set the maximum number of IGMP groups that a Layer 2 interface can join.

IGMP filtering controls only group-specific query and membership reports, including join and leave reports. It does not control general IGMP queries. IGMP filtering has no relationship with the function that directs the forwarding of IP multicast traffic. The filtering feature operates in the same manner whether CGMP or MVR is used to forward the multicast traffic.

IGMP filtering applies only to the dynamic learning of IP multicast group addresses, not static configuration.

With the IGMP throttling feature, you can set the maximum number of IGMP groups that a Layer 2 interface can join. If the maximum number of IGMP groups is set, the IGMP snooping forwarding table contains the maximum number of entries, and the interface receives an IGMP join report, you can configure an interface to drop the IGMP report or to replace the randomly selected multicast entry with the received IGMP report.

## Default IGMP Filtering and Throttling Configuration

The following table displays the default IGMP filtering and throttling configuration for the device.

*Table 6: Default IGMP Filtering Configuration*

| Feature | Default Setting |
|---|---|
| IGMP filters | None applied. |
| IGMP maximum number of IGMP groups | No maximum set.<br>**Note** When the maximum number of groups is in the forwarding table, the default IGMP throttling action is to deny the IGMP report. |
| IGMP profiles | None defined. |
| IGMP profile action | Deny the range addresses. |

# Configuring IGMP Profiles

To configure an IGMP profile, use the **ip igmp profile** global configuration command with a profile number to create an IGMP profile and to enter IGMP profile configuration mode. From this mode, you can specify the parameters of the IGMP profile to be used for filtering IGMP join requests from a port.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **ip igmp profile** *profile-number*<br>**Example:**<br>`Router(config)# ip igmp profile 3` | Enters IGMP profile configuration mode, and assigns a number to the profile you are configuring. The range is from 1 to 4294967295. |
| **Step 3** | **permit** \| **deny**<br>**Example:**<br>`Router(config-igmp-profile)# permit` | (Optional) Sets the action to permit or deny access to the IP multicast address. If no action is configured, the default for the profile is to deny access. |
| **Step 4** | **range** *ip multicast address*<br>**Example:**<br>`Router(config-igmp-profile)# range 229.9.9.0` | Enter the IP multicast address or range of IP multicast addresses to which access is being controlled. If entering a range, enter the low IP multicast address, a space, and the high IP multicast address.<br>You can use the **range** command multiple times to enter multiple addresses or ranges of addresses. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config-igmp-profile)# end` | Returns to privileged EXEC mode. |

# Applying IGMP Profiles

To control access as defined in an IGMP profile, use the ip igmp filter interface configuration command to apply the profile to the appropriate interfaces. You can apply IGMP profiles to layer 2 access ports only; you cannot apply IGMP profiles to routed ports or SVIs. You cannot apply profiles to ports that belong to an EtherChannel port group. You can apply a profile to multiple interfaces, but each interface can only have one profile applied to it.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *interface-id*<br>**Example:**<br>`Router(config)# interface GigabitEthernet0/1/0` | Specifies the physical interface, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group. |
| **Step 3** | **ip igmp filter** *profile-number*<br>**Example:**<br>`Router(config-if)# ip igmp filter 123` | Applies the specified IGMP profile to the interface. The range is 1 to 4294967295. |
| **Step 4** | **end**<br>**Example:**<br>`Router(config-if)# end` | Returns to privileged EXEC mode. |

# Setting the Maximum Number of IGMP Groups

You can set the maximum number of IGMP groups that a Layer 2 interface can join by using the **ip igmp max-groups interface** configuration command. Use the **no** form of this command to set the maximum back to the default, which is no limit. This restriction can be applied to Layer 2 ports only; you cannot set a maximum number of IGMP groups on routed ports or SVIs. You also can use this command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *interface-id*<br>**Example:**<br>`Router(config)# interface GigabitEthernet0/1/0` | Specifies the physical interface to be configured, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group. |
| **Step 3** | **ip igmp max-groups** *number*<br>**Example:**<br>`Router(config-if)# ip igmp max-groups 20` | Sets the maximum number of IGMP groups that the interface can join. The range is 0 to 4294967294. The default is to have no maximum set. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **end**<br>**Example:**<br>`Router(config-if)# end` | Returns to privileged EXEC mode. |

# Configuring the IGMP Throttling Action

After you set the maximum number of IGMP groups that a Layer 2 interface can join, you can configure an interface to remove a randomly selected multicast entry in the forwarding table and to add the next IGMP group to it by using the ip igmp max-groups action replace interface configuration command.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *interface-id*<br>**Example:**<br>`Router(config)# interface GigabitEthernet0/1/0` | Specifies the physical interface to be configured, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group. |
| **Step 3** | **ip igmp max-groups action** {**deny** \| **replace**}<br>**Example:**<br>`Router(config-if)# ip igmp max-groups action replace` | When an interface receives an IGMP report and the maximum number of entries is in the forwarding table, specifies the action that the interface takes:<br>• **deny** —Drop the report.<br>• **replace** —Remove a randomly selected multicast entry in the forwarding table, and add the IGMP group in the report.<br>To prevent the device from removing the forwarding-table entries, you can configure the IGMP throttling action before an interface adds entries to the forwarding table. |
| **Step 4** | **end**<br>**Example:**<br>`Router(config-if)# end` | Returns to privileged EXEC mode. |

# MLD Snooping

In IP Version 4 (IPv4), Layer 2 switches can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on the links that are directly attached to the routers and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD Version 1 (MLDv1) is equivalent to IGMPv2, and MLD Version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol Version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

## MLD Snooping Configuration Guidelines

When configuring MLD snooping, consider these guidelines:

- You can configure MLD snooping characteristics at any time, but you must globally enable MLD snooping by using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping act independently of each other. You can enable both features at the same time on the switch.

## Default MLD Snooping Configuration

*Table 7: Default MLD Snooping Configuration*

| Feature | Default Setting |
|---|---|
| MLD snooping (Global) | Disabled. |
| MLD snooping (per VLAN) | Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place. |
| IPv6 Multicast addresses | None configured. |
| IPv6 Multicast router ports | None configured. |
| MLD snooping Immediate Leave | Disabled. |
| MLD snooping robustness variable | Global: 2; Per VLAN: 0.<br>**Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count. |
| **Note** | The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count. |

| Feature | Default Setting |
|---|---|
| Last listener query count | Global: 2; Per VLAN: 0.<br>**Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count. |
| **Note** | The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count. |
| Last listener query interval | Global: 1000 (1 second); VLAN: 0.<br>**Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval. |
| **Note** | The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval. |
| TCN query solicit | Disabled. |
| TCN query count | 2 |
| MLD listener suppression | |

# Enabling or Disabling MLD Snooping on a VLAN

To enable MLD snooping on a VLAN, perform this procedure:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **ipv6 mld snooping**<br>**Example:**<br>`Router(config)# ipv6 mld snooping` | Enables MLD snooping on the switch. |
| **Step 4** | **ipv6 mld snooping vlan** *vlan-id*<br>**Example:** | Enables MLD snooping on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config)# ipv6 mld snooping vlan 1` | **Note** MLD snooping must be globally enabled for VLAN snooping to be enabled. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

# Configuring UniDirectional Link Detection

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it disables the affected port and alerts you. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

## Enabling UDLD Globally

Follow these steps to enable UDLD in the aggressive or normal mode and to set the configurable message timer on all fiber-optic ports on the device.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **udld** {**aggressive** \| **enable** \| **message time** *message-timer-interval*}<br>**Example:**<br>`Router(config)# udld enable message time 10` | Specifies the UDLD mode of operation:<br>• **aggressive** —Enables UDLD in aggressive mode on all fiber-optic ports.<br>• **enable** —Enables UDLD in normal mode on all fiber-optic ports on the . UDLD is disabled by default.<br>An individual interface configuration overrides the setting of the **udld enable** global configuration command.<br>• **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. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** This command affects fiber-optic ports only. Use the **udld** interface configuration command to enable UDLD on other port types.<br>Use the **no** form of this command, to disable UDLD. |
| **Step 3** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

# Enabling UDLD on an Interface

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **interface** *interface-id*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Specifies the port to be enabled for UDLD, and enters interface configuration mode. |
| **Step 3** | **udld port** [**aggressive**]<br>**Example:**<br>`Router(config-if)# udld port aggressive` | UDLD is disabled by default.<br>• **udld port** —Enables UDLD in normal mode on the specified port.<br>• **udld port aggressive** —(Optional) Enables UDLD in aggressive mode on the specified port.<br>**Note** Use the **no udld port** interface configuration command to disable UDLD on a specified fiber-optic port. |
| **Step 4** | **end**<br>**Example:** | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config-if)# end` | |

# Configuring the Switched Port Analyzer

This section describes how to configure a Switched Port Analyzer (SPAN) session.

- IR8340 can support 66 SPAN sessions in all ports. However, only eight of them can be used as source sessions which includes local SPAN sessions and remote SPAN source sessions. The remaining sessions can be used as remote SPAN destination sessions.
- The session ID range is from 1 to 66.

**Note** Tx, Rx, or both Tx and Rx monitoring is supported.

## SPAN and RSPAN

You can analyze network traffic passing through ports or VLANs by using SPAN or RSPAN to send a copy of the traffic to another port on the device or on another device that has been connected to a network analyzer or other monitoring or security device. SPAN copies (or mirrors) traffic received or sent (or both) on source ports or source VLANs to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports or VLANs. You must dedicate the destination port for SPAN use. Destination ports do not receive or forward traffic by default. It can receive or forward traffic when ingress-forwarding is enabled on the destination ports.

Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another VLAN to the source VLAN cannot be monitored; however, traffic that is received on the source VLAN and routed to another VLAN can be monitored.

You can use the SPAN or RSPAN destination port to inject traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

## Creating a Local SPAN Session

Follow these steps to create a SPAN session and specify the source (monitored) ports or VLANs and the destination (monitoring) ports.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# `**`configure terminal`** | Enter global configuration mode. |
| **Step 3** | **no monitor session** {*session_number* \| **all** \| **local** \| **remote**}<br>**Example:**<br>`Router(config)# `**`no monitor session all`** | Removes any existing SPAN configuration for the session.<br>• For *session_number*, the range is 1 to 66.<br>• **all** —Removes all SPAN sessions.<br>• **local** —Removes all local sessions.<br>• **remote** —Removes all remote SPAN sessions. |
| **Step 4** | **monitor session** *session_number* **source** {**interface** *interface-id* \| **vlan** *vlan-id*} [, \| -] [**both** \| **rx** \| **tx**]<br>**Example:**<br>`Router(config)# `**`monitor session 1 source interface gigabitethernet 0/1/0`** | Specifies the SPAN session and the source port/Vlan (monitored port).<br>• For session_number , the range is 1 to 66.<br>• For interface-id , specify the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** port-channel-number ). Valid port-channel numbers are 1 to 32.<br>• For vlan-id , specify the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN).<br>**Note** A single session can include multiple sources (ports or VLANs) defined in a series of commands, but you cannot combine source ports and source VLANs in one session.<br>• (Optional) [**,** \| **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.<br>• (Optional) **both** \| **rx** \| **tx** —Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic.<br>  • **both** —Monitors both received and sent traffic. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **rx** —Monitors received traffic.<br>• **tx** —Monitors sent traffic.<br>**Note** You can use the **monitor session** session_number **source** command multiple times to configure multiple source ports. |
| **Step 5** | **monitor session** *session_number* **destination** {**interface** *interface-id* [, \| -] [**encapsulation** {**replicate** \| **dot1q**}]}<br>**Example:**<br>`Router(config)#` **`monitor session 1 destination interface gigabitethernet 0/1/0 encapsulation replicate`** | **Note** For local SPAN, you must use the same session number for the source and destination interfaces.<br>• For session_number , specify the session number entered in step 4.<br>• (Optional) [**,** \| **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.<br>(Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).<br>(Optional) **encapsulation dot1q** specifies that the destination interface accepts the source interface incoming packets with IEEE 802.1Q encapsulation.<br>**Note** You can use **monitor session** session_number **destination** command multiple times to configure multiple destination ports. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config)#` **`end`** | Returns to privileged EXEC mode. |
| **Step 7** | **show running-config**<br>**Example:**<br>`Router#` **`show running-config`** | Verifies your entries. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | **copy running-config startup-config**<br>**Example:**<br>`Router# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Creating a Local SPAN with Incoming Traffic Allowed on Destination

Follow these steps to create a SPAN session, to specify the source ports or VLANs and the destination ports, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **no monitor session** {*session_number* \| **all** \| **local** \| **remote**}<br>**Example:**<br>`Router(config)# no monitor session all` | Removes any existing SPAN configuration for the session.<br>• For *session_number*, the range is 1 to 66.<br>• **all** —Removes all SPAN sessions.<br>• **local** —Removes all local sessions.<br>• **remote** —Removes all remote SPAN sessions. |
| **Step 4** | **monitor session** *session_number* **source** {**interface** *interface-id* \| **vlan** *vlan-id*} [, \| -] [**both** \| **rx** \| **tx**]<br>**Example:**<br>`Router(config)# monitor session 2 source gigabitethernet 0/1/0 rx` | Specifies the SPAN session and the source port (monitored port). |
| **Step 5** | **monitor session** *session_number* **destination** {**interface** *interface-id* [, \| -] [**encapsulation** {**replicate**] [**ingress** {**dot1q vlan** *vlan-id* \| **untagged vlan** *vlan-id* \| **vlan** *vlan-id*}]}<br>**Example:** | Specifies the SPAN session, the destination port, the packet encapsulation, and the ingress VLAN and encapsulation.<br>• For session_number , specify the session number entered in Step 4. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config)#` **`monitor session 2 destination interface gigabitethernet 0/1/0 encapsulation replicate ingress dot1q vlan 6`** | • For interface-id , specify the destination port. The destination interface must be a physical port or port-channel; it cannot be an EtherChannel, and it cannot be a VLAN.<br>• (Optional) [**,** \| **-**] —Specifies a series or range of interfaces. Enter a space before and after the comma or hyphen.<br>• (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).<br>• (Optional) **encapsulation dot1q** specifies that the destination interface accepts the source interface incoming packets with IEEE 802.1Q encapsulation.<br>• **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:<br>  • **dot1q vlan** vlan-id— Accepts incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN.<br>  • **untagged vlan** vlan-id or **vlan** vlan-id— Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config)#` **`end`** | Returns to privileged EXEC mode. |
| **Step 7** | **show running-config**<br>**Example:**<br>`Router#` **`show running-config`** | Verifies your entries. |
| **Step 8** | **copy running-config startup-config**<br>**Example:**<br>`Router#` **`copy running-config startup-config`** | (Optional) Saves your entries in the configuration file. |

# Specifying VLANs to Filter

Follow these steps to limit SPAN source traffic to specific VLANs.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>- Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **no monitor session** {*session_number* \| **all** \| **local** \| **remote**}<br>**Example:**<br>`Router(config)# no monitor session all` | Removes any existing SPAN configuration for the session.<br>- For *session_number*, the range is 1 to 66.<br>- **all** —Removes all SPAN sessions.<br>- **local** —Removes all local sessions.<br>- **remote** —Removes all remote SPAN sessions. |
| **Step 4** | **monitor session** *session_number* **source** {**interface** *interface-id*]<br>**Example:**<br>`Router(config)# monitor session 2 source interface gigabitethernet 0/1/0 rx` | Specifies the characteristics of the source port (monitored port) and SPAN session.<br>- For session_number , the range is 1 to 66.<br>- For interface-id , specify the source port to monitor. The interface specified must already be configured as a trunk port. |
| **Step 5** | **monitor session** *session_number* **filter vlan** *vlan-id* [, \| -]<br>**Example:**<br>`Router(config)# monitor session 2 filter vlan 1 - 5 , 9` | Limits the SPAN source traffic to specific VLANs.<br>- For session_number , enter the session number specified in Step 4.<br>- For vlan-id , the range is 1 to 4094.<br>- (Optional) Use a comma (**,** ) to specify a series of VLANs, or use a hyphen (**-** ) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **monitor session** *session_number* **destination** {**interface** *interface-id* [, \| -] [**encapsulation replicate** \| **encapsulation dot1q**}]}<br>**Example:**<br>`Router(config)# monitor session 2 destination interface gigabitethernet 0/1/0` | Specifies the SPAN session and the destination port (monitoring port).<br>• For session_number , specify the session number entered in Step 4.<br>• For interface-id , specify the destination port. The destination interface must be a physical port or port-channel; it cannot be an EtherChannel, and it cannot be a VLAN.<br>• (Optional) [**,** \| **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.<br>• (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).<br>• (Optional) **encapsulation dot1q** IEEE 802.1Q is a standard protocol for interconnecting multiple switches and routers and for defining VLAN topologies. Applies a VLAN ID to the subinterface. |
| **Step 7** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| **Step 8** | **show running-config**<br>**Example:**<br>`Router# show running-config` | Verifies your entries. |
| **Step 9** | **copy running-config startup-config**<br>**Example:**<br>`Router# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Verifying the SPAN Session

Use the **show monitor session** command to verify the sources and destinations configured for the SPAN session.

```
Router#show monitor session 1

Session 1
---------
Session 1
---------
Type : Local Session
Source Ports :
Both : Gi0/1/0
Destination Ports : Gi0/1/1
```

# Removing a SPAN Session

To remove sources or destinations from the SPAN session, use the **no monitor session** session command in global configuration mode as shown in the following example:

```
Router(config)#no monitor session 1
```

# Configuring a VLAN as an RSPAN VLAN

Follow these steps to create a new VLAN, then configure it to be the RSPAN VLAN for the RSPAN session.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **vlan** *vlan-id*<br>**Example:**<br>`Router(config)# vlan 100` | Enters a VLAN ID to create a VLAN, or enters the VLAN ID of an existing VLAN, and enters VLAN configuration mode. The range is 2 to 1001 and 1006 to 4094.<br>The RSPAN VLAN cannot be VLAN 1 (the default VLAN) or VLAN IDs 1002 through 1005 (reserved for Token Ring and FDDI VLANs). |
| **Step 4** | **remote-span**<br>**Example:**<br>`Router(config)# remote-span` | Configures the VLAN as an RSPAN VLAN. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **show running-config**<br>**Example:**<br>`Router# show running-config` | Verifies your entries. |
| **Step 7** | **copy running-config startup-config**<br>**Example:**<br>`Router# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

### What to do next

You must create the RSPAN VLAN in all devices that will participate in RSPAN. If the RSPAN VLAN-ID is in the normal range (lower than 1005) and VTP is enabled in the network, you can create the RSPAN VLAN in one device, and VTP propagates it to the other devices in the VTP domain. For extended-range VLANs (greater than 1005), you must configure RSPAN VLAN on both source and destination devices and any intermediate devices.

Use VTP pruning to get an efficient flow of RSPAN traffic, or manually delete the RSPAN VLAN from all trunks that do not need to carry the RSPAN traffic.

To remove the remote SPAN characteristic from a VLAN and convert it back to a normal VLAN, use the **no remote-span** VLAN configuration command.

To remove a source port or VLAN from the SPAN session, use the **no monitor session** session_number **source** {**interface** interface-id | **vlan** vlan-id} global configuration command. To remove the RSPAN VLAN from the session, use the **no monitor session** session_number {**Source|destination**} **remote vlan** vlan-id .

# Creating an RSPAN Source Session

Follow these steps to create and start an RSPAN source session and to specify the monitored source and the destination RSPAN VLAN.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **no monitor session** {*session_number* \| **all** \| **local** \| **remote**}<br>**Example:**<br>`Router(config)# no monitor session all` | Removes any existing SPAN configuration for the session.<br>• For *session_number*, the range is 1 to 66. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **all** —Removes all SPAN sessions.<br>• **local** —Removes all local sessions.<br>• **remote** —Removes all remote SPAN sessions. |
| **Step 4** | **monitor session** *session_number* **source** {**interface** *interface-id* \| **vlan** *vlan-id*} [, \| -] [**both** \| **rx** \| **tx**]<br>**Example:**<br>`Router(config)# monitor session 1 source interface gigabitethernet 0/1/0 tx` | Specifies the RSPAN session and the source port (monitored port).<br>• For session_number , the range is 1 to 66.<br>• Enter a source port or source VLAN for the RSPAN session:<br>  • For interface-id , specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** port-channel-number ). Valid port-channel numbers are 1 to 32.<br>  • For vlan-id , specifies the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN).<br>  A single session can include multiple sources (ports or VLANs), defined in a series of commands, but you cannot combine source ports and source VLANs in one session.<br>• (Optional) [**,** \| **-**] —Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.<br>• (Optional) **both** \| **rx** \| **tx** —Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic.<br>  • **both** —Monitors both received and sent traffic.<br>  • **rx** —Monitors received traffic.<br>  • **tx** —Monitors sent traffic. |
| **Step 5** | **monitor session** *session_number* **destination remote vlan** *vlan_id* | Specifies the RSPAN session, the destination RSPAN VLAN, and the destination-port group. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router(config)# monitor session 1 destination remote vlan 100` | - For session_number , enter the number defined in Step 4.<br>- For vlan-id , specify the RSPAN VLAN in source session, which will transport mirrored traffic to destination session. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| **Step 7** | **show running-config**<br>**Example:**<br>`Router# show running-config` | Verifies your entries. |
| **Step 8** | **copy running-config startup-config**<br>**Example:**<br>`Router# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Specifying VLANs to Filter on RSPAN Source Session

Follow these steps to configure the RSPAN source session to limit RSPAN source traffic to specific VLANs.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>- Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **no monitor session** {*session_number* \| **all** \| **local** \| **remote**}<br>**Example:**<br>`Router(config)# no monitor session 2` | Removes any existing SPAN configuration for the session.<br>- For *session_number*, the range is 1 to 66.<br>- **all** —Removes all SPAN sessions.<br>- **local** —Removes all local sessions.<br>- **remote** —Removes all remote SPAN sessions. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **monitor session** *session_number* **source interface** *interface-id*<br>**Example:**<br>`Router(config)# monitor session 2 source interface gigabitethernet 0/1/0 rx` | Specifies the characteristics of the source port (monitored port) and SPAN session.<br>• For session_number , the range is 1 to 66.<br>• For interface-id , specify the source port to monitor. The interface specified must already be configured as a trunk port. |
| **Step 5** | **monitor session** *session_number* **filter vlan** *vlan-id* [, \| -]<br>**Example:**<br>`Router(config)# monitor session 2 filter vlan 1 - 5 , 9` | Limits the SPAN source traffic to specific VLANs.<br>• For session_number , enter the session number specified in step 4.<br>• For vlan-id , the range is 1 to 4094.<br>• (Optional) **,** \| **-** Use a comma (**,** ) to specify a series of VLANs or use a hyphen (**-** ) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen. |
| **Step 6** | **monitor session** *session_number* **destination remote vlan** *vlan-id*<br>**Example:**<br>`Router(config)# monitor session 2 destination remote vlan 902` | Specifies the RSPAN session and the destination remote VLAN (RSPAN VLAN).<br>• For session_number , enter the session number specified in Step 4.<br>• For vlan-id , specify the RSPAN VLAN to carry the monitored traffic to the destination port. |
| **Step 7** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| **Step 8** | **show running-config**<br>**Example:**<br>`Router# show running-config` | Verifies your entries. |
| **Step 9** | **copy running-config startup-config**<br>**Example:**<br>`Router# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Creating an RSPAN Destination Session and Configuring Incoming Traffic

Follow these steps to create an RSPAN destination session, to specify the source RSPAN VLAN and the destination port, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>- Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **no monitor session** {*session_number* \| **all** \| **local** \| **remote**}<br>**Example:**<br>`Router(config)# no monitor session 2` | Removes any existing SPAN configuration for the session.<br>- For *session_number*, the range is 1 to 66.<br>- **all** —Removes all SPAN sessions.<br>- **local** —Removes all local sessions.<br>- **remote** —Removes all remote SPAN sessions. |
| **Step 4** | **monitor session** *session_number* **source remote vlan** *vlan-id*<br>**Example:**<br>`Router(config)# monitor session 2 source remote vlan 901` | Specifies the RSPAN session and the source RSPAN VLAN.<br>- For session_number , the range is 1 to 66.<br>- For vlan-id , specify the RSPAN VLAN in destination session, which will receive mirrored traffic from the source session. |
| **Step 5** | **monitor session** *session_number* **destination** {**interface** *interface-id* [, \| -] [**ingress** {**dot1q vlan** *vlan-id* \| **untagged vlan** *vlan-id* \| **vlan** *vlan-id*}]}<br>**Example:**<br>`Router(config)# monitor session 2 destination interface gigabitethernet 0/1/0 ingress vlan 6` | Specifies the SPAN session, the destination port, the packet encapsulation, and the incoming VLAN and encapsulation.<br>- For session_number , enter the number defined in Step 5.<br>In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port.<br>- For interface-id , specify the destination interface. The destination interface must be a physical interface. |

| | Command or Action | Purpose |
|---|---|---|
| | | • Though visible in the command-line help string, **encapsulation replicate** is not supported for RSPAN. The original VLAN ID is overwritten by the RSPAN VLAN ID, and all packets appear on the destination port as untagged.<br>• (Optional) [**,** \| **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.<br>• Enter **ingress** with additional keywords to enable forwarding of incoming traffic on the destination port and to specify the encapsulation type:<br>  • **dot1q vlan** vlan-id— Forwards incoming packets with IEEE 802.1Q encapsulation with the specified VLAN as the default VLAN.<br>  • **untagged vlan** vlan-id or **vlan** vlan-id— Forwards incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| **Step 7** | **show running-config**<br>**Example:**<br>`Router# show running-config` | Verifies your entries. |
| **Step 8** | **copy running-config startup-config**<br>**Example:**<br>`Router# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# EtherChannel Overview

EtherChannel provides fault-tolerant high-speed links between switches, routers, and servers. You can use the EtherChannel to increase the bandwidth between the wiring closets and the data center, and you can deploy it anywhere in the network where bottlenecks are likely to occur. 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.

An EtherChannel consists of individual Ethernet links bundled into a single logical link

The EtherChannel provides full-duplex bandwidth up to 4 Gb/s (Gigabit EtherChannel) between your switch and another switch or host.

Each EtherChannel can consist of up to four compatibly configured Ethernet ports.

## Channel Groups and Port-Channel Interfaces

An EtherChannel comprises a channel group and a port-channel interface. The channel group binds physical ports to the port-channel interface. Configuration changes applied to the port-channel interface apply to all the physical ports bound together in the channel group. 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 32. This port-channel interface number corresponds to the one specified with the channel-group interface configuration command.

## Port Aggregation Protocol

The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco devices and on those devices licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports.

By using PAgP, the device learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports (on a single device in the stack) into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single device port.

## Link Aggregation Control Protocol

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

By using LACP, the switch learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, LACP adds the group to the spanning tree as a single device port.

## Configuring Layer 2 EtherChannels

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

Use the **show etherchannel swport xxx** command to view the EtherChannels.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **interface** [*interface-id*]<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Specifies a physical port, and enters interface configuration mode.<br>Valid interfaces are physical ports.<br>For a PAgP EtherChannel, you can configure up to four ports of the same type and speed for the same group.<br>For a LACP EtherChannel, you can configure up to 8 Ethernet ports of the same type. Up to eight ports can be active, and up to eight ports can be in standby mode. |
| **Step 4** | **switchport mode** {**access** \| **trunk**}<br>**Example:**<br>`Router(config-if)# switchport mode access` | Assigns all ports as static-access ports in the same VLAN, or configure them as trunks. If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094. |
| **Step 5** | **switchport access vlan** *vlan-id*<br>**Example:**<br>`Router(config-if)# switchport access vlan 22` | (Optional) If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094. |
| **Step 6** | **channel-group channel-group-number mode {auto [non-silent] \| desirable [non-silent ] \| on } \| { active \| passive}**<br>**Example:**<br>`Router(config-if)# channel-group 5 mode auto` | Assigns the port to a channel group, and specifies the PAgP or the LACP mode.<br>For **mode** , select one of these keywords:<br>• **auto —** Enables PAgP only if a PAgP device is detected. It places the port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not start PAgP packet negotiation.<br>• **desirable —** Unconditionally enables PAgP. It places the port into an active negotiating state, in which the port starts negotiations with other ports by sending PAgP packets. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **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.<br>• **non-silent —** (Optional) If your device is connected to a partner that is PAgP-capable, configures the device port for nonsilent operation when the port is in the **auto** or **desirable** mode. If you do not specify **non-silent** , silent is assumed. The silent setting is for connections to file servers or packet analyzers. This setting allows PAgP to operate, to attach the port to a channel group, and to use the port for transmission.<br>• **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.<br>• **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. |
| **Step 7** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

# Configuring EtherChannel Load-Balancing

You can configure EtherChannel load-balancing to use one of several different forwarding methods.

This task is optional.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **port-channel swport load-balance { dst-ip \| dst-mac \| dst-mixed-ip-port \| dst-port \| extended [ dst-ip \| dst-mac \| dst-port \| ipv6-label \| l3-proto \| src-ip \| src-mac \| src-port ] \| src-dst-ip \| src-dst-mac src-dst-mixed-ip-port src-dst-portsrc-ip \| src-mac \| src-mixed-ip-port \| src-port }**<br>**Example:**<br>`Router(config)# **port-channel swport load-balance src-mac**` | Configures an EtherChannel load-balancing method.<br>Select one of these load-distribution methods:<br>• **dst-ip** —Specifies destination-host IP address.<br>• **dst-mac** —Specifies the destination-host MAC address of the incoming packet.<br>• **dst-mixed-ip-port** —Specifies the host IP address and TCP/UDP port.<br>• **dst-port** —Specifies the destination TCP/UDP port.<br>• **extended** —Specifies extended load balance methods--combinations of source and destination methods beyond those available with the standard command.<br>• **ipv6-label** —Specifies the IPv6 flow label.<br>• **l3-proto** —Specifies the Layer 3 protocol.<br>• **src-dst-ip** —Specifies the source and destination host IP address.<br>• **src-dst-mac** —Specifies the source and destination host MAC address.<br>• **src-dst-mixed-ip-port** —Specifies the source and destination host IP address and TCP/UDP port.<br>• **src-dst-port** —Specifies the source and destination TCP/UDP port.<br>• **src-ip** —Specifies the source host IP address.<br>• **src-mac** —Specifies the source MAC address of the incoming packet.<br>• **src-mixed-ip-port** —Specifies the source host IP address and TCP/UDP port.<br>• **src-port** —Specifies the source TCP/UDP port. |
| **Step 3** | **end**<br>**Example:**<br>`Router(config)# **end**` | Returns to privileged EXEC mode. |

# Configuring the PAgP Learn Method and Priority

This task is optional.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **interface** [*interface-id*]<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Specifies the port for transmission, and enters interface configuration mode. |
| **Step 4** | **pagp learn-method physical-port**<br>**Example:**<br>`Router(config-if)# pagp learn-method physical port` | Selects the PAgP learning method.<br>By default, **aggregation-port learning** is selected, which means the device sends packets to the source by using any of the ports in the EtherChannel. With aggregate-port learning, it is not important on which physical port the packet arrives.<br>Selects **physical-port** to connect with another device that `is` a physical learner.<br>Make sure to configure the **port-channel load-balance** global configuration command to **src-mac** .<br>The learning method must be configured the same at both ends of the link. |
| **Step 5** | **pagp port-priority priority**<br>**Example:**<br>`Router(config-if)# pagp port-priority 200` | Assigns a priority so that the selected port is chosen for packet transmission.<br>For *priority*, the range is 0 to 255. The default is 128. The higher the priority, the more likely that the port will be used for PAgP transmission. |
| **Step 6** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

## Configuring the LACP Port Channel Min-Links Feature

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.

To configure the minimum number of links that are required for a port channel. Perform the following tasks.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 3** | **interface port-channel** [*channel-number*]<br>**Example:**<br>`Router(config)# interface port-channel 2` | Enters interface configuration mode for a port-channel.<br>For *channel-number*, the range is 1 to 63. |
| **Step 4** | **port-channel min-links** *min-links-number*<br>**Example:**<br>`Router(config-if)# port-channel min-links 3` | 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.<br>For *min-links-number* , the range is 2 to 8. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

## Configuring LACP Fast Rate Timer

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lacp rate** command to set the rate at which LACP control packets are received by an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable** | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router> enable` | • Enter your password if prompted. |
| Step 2 | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| Step 3 | **interface** *type slot/port*<br>**Example:**<br>`Router(config)# interface gigabitEthernet 0/1/0` | Configures an interface and enters interface configuration mode. |
| Step 4 | **lacp rate { normal \| fast}**<br>**Example:**<br>`Router(config-if)# lacp rate fast` | Configures the rate at which LACP control packets are received by an LACP-supported interface.<br>To reset the timeout rate to its default, use the **no lacp rate** command. |
| Step 5 | **end**<br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |
| Step 6 | **show lacp internal**<br>**Example:**<br>`Router# show lacp internal`<br>`Router# show lacp counters` | Verifies your configuration. |

# Modular Quality of Service Command-Line Interface

The MQC (Modular Quality of Service (QoS) Command-Line Interface (CLI)) enables you to set packet classification and marking based on a QoS group value. ith the device, QoS features are enabled through the Modular QoS command-line interface (MQC). The MQC is a command-line interface (CLI) structure that allows you to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to classify traffic, while the QoS features in the traffic policy determine how to treat the classified traffic. One of the main goals of MQC is to provide a platform-independent interface for configuring QoS across Cisco platforms. For more infomraton on the Modular Quality of Service, see the Quality of Service Configuration Guide, Cisco IOS XE Fuji 16.9.x.

## Creating a Traffic Class

To create a traffic class containing match criteria, use the **class-map** command to specify the traffic class name, and then use the following **match** commands in class-map configuration mode, as needed.

### Before you begin

All match commands specified in this configuration task are considered optional, but you must configure at least one match criterion for a class.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router#` **`configure terminal`** | Enter global configuration mode. |
| **Step 2** | **class-map** *class-map name* {**match-any**}<br>**Example:**<br>`Router(config)#` **`class-map type ngsw-qos test_1000`**<br>`Router(config-cmap)#` | Enters class map configuration mode.<br>• Creates a class map to be used for matching packets to the class whose name you specify.<br>• **match-any:** Any one of the match criteria must be met for traffic entering the traffic class to be classified as part of it. |
| **Step 3** | **match access-group** { *index number* \| *name*}<br>**Example:**<br>`Router(config-cmap)#` **`match access-group 100`**<br>`Router(config-cmap)#` | The following parameters are available for this command:<br>• access-group<br>• cos<br>• dscp<br>• group-object<br>• ip<br>• mpls<br>• precedence<br>• protocol<br>• qos-group<br>• vlan<br>• wlan<br>(Optional) For this example, enter the access-group ID:<br>• Access list index (value from 1 to 2799)<br>• Named access list |
| **Step 4** | **match cos** *cos value*<br>**Example:** | (Optional) Matches IEEE 802.1Q or ISL class of service (user) priority values. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config-cmap)# match cos 2 3 4 5` | • Enters up to 4 CoS values separated by spaces (0 to 7). |
| **Step 5** | **match dscp** *dscp value*<br>**Example:**<br>`Router(config-cmap)# match dscp af11 af12` | (Optional) Matches the DSCP values in IPv4 and IPv6 packets. |
| **Step 6** | **match ip** { **dscp** *dscp value* \| **precedence** *precedence value* }<br>**Example:**<br>`Router(config-cmap)# match ip dscp af11 af12` | (Optional) Matches IP values including the following:<br>• **dscp**—Matches IP DSCP (DiffServ codepoints).<br>• **precedence**—Matches IP precedence (0 to 7). |
| **Step 7** | **match qos-group** *qos group value*<br>**Example:**<br>`Router(config-cmap)# match qos-group 10` | (Optional) Matches QoS group value (from 0 to 31). |
| **Step 8** | **match vlan** *vlan value*<br>**Example:**<br>`Router(config-cmap)# match vlan 210` | (Optional) Matches a VLAN ID (from 1 to 4095). |
| **Step 9** | **end**<br>**Example:**<br>`Router(config)# end` | Saves the configuration changes. |

### What to do next

Configure the policy map.

# Creating a Traffic Policy

To create a traffic policy, use the **policy-map** global configuration command to specify the traffic policy name.

The traffic class is associated with the traffic policy when the **class** command is used. The **class** command must be entered after you enter the policy map configuration mode. After entering the **class** command, the device is automatically in policy map class configuration mode, which is where the QoS policies for the traffic policy are defined.

The following policy map class-actions are supported:

- bandwidth—Bandwidth configuration options.
- exit—Exits from the QoS class action configuration mode.
- no—Negates or sets default values for the command.

- police—Policer configuration options.
- priority—Strict scheduling priority configuration options for this class.
- queue-buffers—Queue buffer configuration options.
- queue-limit—Queue maximum threshold for Weighted Tail Drop (WTD) configuration options.
- service-policy—Configures the QoS service policy.
- set—Sets QoS values using the following options:
    - CoS values
    - DSCP values
    - Precedence values
    - QoS group values
- shape—Traffic-shaping configuration options.

### Before you begin

You should have first created a class map.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **policy-map type** *policy-map name*<br>**Example:**<br>`Router(config)# policy-map type ngsw-qos test_1000` | Enters policy map configuration mode.<br>Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| **Step 3** | **class** { *class-name* \| **class-default**}<br>**Example:**<br>`Router(config-pmap)# class test_1000` | Specifies the name of the class whose policy you want to create or change.<br>You can also create a system default class for unclassified packets. |
| **Step 4** | **bandwidth** { **kb/s** *kb/s value* \| **percent** *percentage* \| **remaining** {*percent* \| *ratio*}}<br>**Example:**<br>`Router(config-pmap-c)# bandwidth 50` | (Optional) Sets the bandwidth using one of the following:<br>• **kb/s**—Kilobits per second, enter a value between 20000 and 10000000 for Kb/s.<br>• **percent**—Enter the percentage of the total bandwidth to be used for this policy map. |

| | Command or Action | Purpose |
|---|---|---|
| | | • **remaining**—Enter the percentage ratio of the remaining bandwidth. |
| **Step 5** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | (Optional) Exits from QoS class action configuration mode. |
| **Step 6** | **no**<br>**Example:**<br>`Router(config-pmap-c)# no` | (Optional) Negates the command. |
| **Step 7** | **police** { *target_bit_rate* \| **cir** \| **rate**}<br>**Example:**<br>`Router(config-pmap-c)# police 100000` | (Optional) Configures the policer:<br>• *target_bit_rate*—Enter the bit rate per second, enter a value between 8000 and 10000000000.<br>• **cir**—Committed Information Rate<br>• **rate**—Specify police rate, PCR for hierarchical policies or SCR for single-level ATM 4.0 policer policies. |
| **Step 8** | **priority level** *level value*<br>**Example:**<br>`Router(config-pmap-c)# priority level 1` | (Optional) Sets the strict scheduling priority for this class. Command options include:<br>• **level**—Establishes a multi-level priority queue. Enter a value (1 or 2). |
| **Step 9** | **queue-buffers ratio** *ratio limit*<br>**Example:**<br>`Router(config-pmap-c)# queue-buffers ratio 10` | (Optional) Configures the queue buffer for the class. Enter the queue buffers ratio limit (0 to 100). |
| **Step 10** | **queue-limit { packets \| cos \| dscp \| percent}**<br>**Example:**<br>`Router(config-pmap-c)# queue-limit cos 7 percent 50` | (Optional) Specifies the queue maximum threshold for the tail drop:<br>• *packets*—Packets by default, enter a value between 1 to 2000000.<br>• **cos**—Enter the parameters for each COS value.<br>• **dscp**—Enter the parameters for each DSCP value.<br>• **percent**—Enter the percentage for the threshold. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 11** | **service-policy** *policy-map name*<br>**Example:**<br>`Router(config-pmap-c)# service-policy test_2000` | (Optional) Configures the QoS service policy. |
| **Step 12** | **set { cos \| dscp \| ip \| precedence \| qos-group \| wlan}**<br>**Example:**<br>`Router(config-pmap-c)# set cos 7` | (Optional) Sets the QoS values. Possible QoS configuration values include:<br>• **cos**—Sets the IEEE 802.1Q/ISL class of service/user priority.<br>• **dscp**—Sets DSCP in IP(v4) and IPv6 packets.<br>• **ip**—Sets IP specific values.<br>• **precedence**—Sets precedence in IP(v4) and IPv6 packet.<br>• **qos-group**—Sets the QoS Group. |
| **Step 13** | **shape average** { *target _bit_rate* \| **percent**}<br>**Example:**<br>`Router(config-pmap-c)# shape average percent 50` | (Optional) Sets the traffic shaping. Command parameters include:<br>• *target_bit_rate*—Target bit rate.<br>• **percent**—Percentage of interface bandwidth for Committed Information Rate. |
| **Step 14** | **end**<br>**Example:**<br>`Router(config)# end` | Saves the configuration changes. |

### What to do next

Configure the interface.

## Configuring Class-Based Packet Marking

This is an important procedure that explains how to configure the following class-based packet marking features on your device:

- CoS value
- DSCP value
- IP value
- Precedence value
- QoS group value

- WLAN value

### Before you begin

You should have created a class map and a policy map before beginning this procedure.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **policy-map type** *policy-map name*<br>**Example:**<br>`Router(config)# policy-map type ngsw-qos policy1`<br>`Device(config-pmap)#` | Enters policy map configuration mode.<br>Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| **Step 3** | **class** *class name*<br>**Example:**<br>`Router(config)# class class1`<br>`Device(config-pmap)#` | Enters policy class map configuration mode. Specifies the name of the class whose policy you want to create or change.<br>Command options for policy class map configuration mode include the following:<br>• **bandwidth**—Bandwidth configuration options.<br>• **exit**—Exits from the QoS class action configuration mode.<br>• **no**—Negates or sets default values for the command.<br>• **police**—Policer configuration options.<br>• **priority**—Strict scheduling priority configuration options for this class.<br>• **queue-buffers**—Queue buffer configuration options.<br>• **queue-limit**—Queue maximum threshold for Weighted Tail Drop (WTD) configuration options.<br>• **service-policy**—Configures the QoS service policy.<br>• **set**—Sets QoS values using the following options:<br>  • CoS values |

| | Command or Action | Purpose |
|---|---|---|
| | | • DSCP values<br>• Precedence values<br>• QoS group values<br>• WLAN values<br>• **shape**—Traffic-shaping configuration options.<br>**Note** This procedure describes the available configurations using **set** command options. The other command options (**bandwidth** ) are described in other sections of this guide. Although this task lists all of the possible **set** commands, only one **set** command is supported per class. |
| **Step 4** | **set cos** {*cos value* \| **cos table** *table-map name* \| **dscp table** *table-map name* \| **precedence table** *table-map name* \| **qos-group table** *table-map name* \| **wlan user-priority table** *table-map name*}<br>**Example:**<br>`Router(config-pmap)# set cos 5` | (Optional) Sets the specific IEEE 802.1Q Layer 2 CoS value of an outgoing packet. Values are from 0 to7.<br>You can also set the following values using the **set cos** command:<br>• **cos table**—Sets the CoS value based on a table map.<br>• **dscp table**—Sets the code point value based on a table map.<br>• **precedence table**—Sets the code point value based on a table map.<br>• **qos-group table**—Sets the CoS value from QoS group based on a table map.<br>• **wlan user-priority table**—Sets the CoS value from the WLAN user priority based on a table map. |
| **Step 5** | **set dhcp** {*dhcp value* \| **default** \| **dscp table** *table-map name* \| **ef** \| **precedence table** *table-map name* \| **qos-group table** *table-map name* \| **wlan user-priority table** *table-map name*}<br>**Example:**<br>`Router(config-pmap)# set dscp af11` | (Optional) Sets the DSCP value.<br>In addition to setting specific DSCP values, you can also set the following using the **set dscp** command:<br>• **default**—Matches packets with default DSCP value (000000). |

| | Command or Action | Purpose |
|---|---|---|
| | | • **dscp table**—Sets the packet DSCP value from DSCP based on a table map.<br>• **ef**—Matches packets with EF DSCP value (101110).<br>• **precedence table**—Sets the packet DSCP value from precedence based on a table map.<br>• **qos-group table**—Sets the packet DSCP value from a QoS group based upon a table map.<br>• **wlan user-priority table**—Sets the packet DSCP value based upon a WLAN user-priority based upon a table map. |
| **Step 6** | **set ip {dscp \| precedence}**<br>**Example:**<br>Router(config-pmap)# **set ip dscp c3** | (Optional) Sets IP specific values. These values are either IP DSCP or IP precedence values.<br>You can set the following values using the **set ip dscp** command:<br>• *dscp value*—Sets a specific DSCP value.<br>• **default**—Matches packets with default DSCP value (000000).<br>• **dscp table**—Sets the packet DSCP value from DSCP based on a table map.<br>• **ef**—Matches packets with EF DSCP value (101110).<br>• **precedence table**—Sets the packet DSCP value from precedence based on a table map.<br>• **qos-group table**—Sets the packet DSCP value from a QoS group based upon a table map.<br>• **wlan user-priority table**—Sets the packet DSCP value based upon a WLAN user-priority based upon a table map.<br>You can set the following values using the **set ip precedence** command:<br>• *precedence value*—Sets the precedence value (from 0 to 7) . |

| | Command or Action | Purpose |
|---|---|---|
| | | • **cos table**—Sets the packet precedence value from Layer 2 CoS based on a table map.<br>• **dscp table**—Sets the packet precedence from DSCP value based on a table map.<br>• **precedence table**—Sets the precedence value from precedence based on a table map<br>• **qos-group table**—Sets the precedence value from a QoS group based upon a table map. |
| **Step 7** | **set precedence** {*precedence value* \| **cos table** *table-map name* \| **dscp table** *table-map name* \| **precedence table** *table-map name* \| **qos-group table** *table-map name*}<br>**Example:**<br>`Router(config-pmap)# set precedence 5` | (Optional) Sets precedence values in IPv4 and IPv6 packets.<br>You can set the following values using the **set precedence** command:<br>• *precedence value*—Sets the precedence value (from 0 to 7) .<br>• **cos table**—Sets the packet precedence value from Layer 2 CoS on a table map.<br>• **dscp table**—Sets the packet precedence from DSCP value on a table map.<br>• **precedence table**—Sets the precedence value from precedence based on a table map.<br>• **qos-group table**—Sets the precedence value from a QoS group based upon a table map. |
| **Step 8** | **set qos-group** {*qos-group value* \| **dscp table** *table-map name* \| **precedence table** *table-map name*}<br>**Example:**<br>`Router(config-pmap)# set qos-group 10` | (Optional) Sets QoS group values. You can set the following values using this command:<br>• *qos-group value*—A number from 1 to 31.<br>• **dscp table**—Sets the code point value from DSCP based on a table map.<br>• **precedence table**—Sets the code point value from precedence based on a table map. |
| **Step 9** | **set wlan user-priority table** {*wlan user-priority table value* \| **cos table** *table-map name* \| **dscp table** *table-map name* \| **qos-group** | (Optional) Sets the WLAN user priority value. You can set the following values using this command: |

| | Command or Action | Purpose |
|---|---|---|
| | **table** *table-map name* \| **wlan table** *table-map name*}<br>**Example:**<br>`Router(config-pmap)# set wlan user-priority 1` | • *wlan user-priority value*—A value between 0 to 7.<br>• **cos table**—Sets the WLAN user priority value from CoS based on a table map.<br>• **dscp table**—Sets the WLAN user priority value from DSCP based on a table map.<br>• **qos-group table**—Sets the WLAN user priority value from QoS group based on a table map.<br>• **wlan table**—Sets the WLAN user priority value from the WLAN user priority based on a table map. |
| **Step 10** | **end**<br>**Example:**<br>`Router(config)# end` | Saves the configuration changes. |
| **Step 11** | **show policy-map**<br>**Example:**<br>`Router(config)# show policy-map` | (Optional) Displays policy configuration information for all classes configured for all service policies. |

### What to do next

Attach the traffic policy to an interface using the **service-policy** command.

# Attaching a Traffic Policy to an Interface

After the traffic class and traffic policy are created, you must use the **service-policy** interface configuration command to attach a traffic policy to an interface, and to specify the direction in which the policy should be applied (either on packets coming into the interface or packets leaving the interface).

### Before you begin

A traffic class and traffic policy must be created before attaching a traffic policy to an interface.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **interface** *type* | |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **service-policy** {**input** *policy-map* \| **output** *policy-map*}<br>**Example:**<br>`Router(config-if)# service-policy output policy_map_01` | Attaches a policy map to an input or output interface. This policy map is then used as the service policy for that interface.<br>In this example, the traffic policy evaluates all traffic leaving that interface. |
| **Step 4** | **end**<br>**Example:**<br>`Router(config)# end` | Saves the configuration changes. |
| **Step 5** | **show policy map**<br>**Example:**<br>`Router(config)# show policy map` | (Optional) Displays statistics for the policy on the specified interface. |

### Example

### What to do next

Proceed to attach any other traffic policy to an interface, and to specify the direction in which the policy should be applied.

## Classifying, Policing, and Marking Traffic on Physical Ports by Using Policy Maps

You can configure a nonhierarchical policy map on a physical port that specifies which traffic class to act on. Actions supported are remarking and policing.

### Before you begin

You should have already decided upon the classification, policing, and marking of your network traffic by policy maps prior to beginning this procedure.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enter global configuration mode. |
| **Step 2** | **class-map** { *class-map name* \| **match-any**}<br>**Example:**<br>`Device(config)# class-map ipclass1`<br>`Device(config-cmap)# exit` | Enters class map configuration mode.<br>• Creates a class map to be used for matching packets to the class whose name you specify. |

| | Command or Action | Purpose |
|---|---|---|
| | | • If you specify **match-any**, one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class. This is the default. |
| **Step 3** | **match access-group** { *access list index* \| *access list name* }<br>**Example:**<br>`Device(config-cmap)# `**`match access-group 1000`**<br>`Device(config-cmap)# `**`exit`** | The following parameters are available for this command:<br>• access-group<br>• cos<br>• dscp<br>• group-object<br>• ip<br>• mpls<br>• precedence<br>• protocol<br>• qos-group<br>• vlan<br>• wlan<br>(Optional) For this example, enter the access-group ID:<br>• Access list index (value from 1 to 2799)<br>• Named access list |
| **Step 4** | **policy-map** *policy-map name*<br>**Example:**<br>`Router(config)# `**`policy-map type ngsw-qos flowit`**<br>`Device(config-pmap)#` | Creates a policy map by entering the policy map name, and enters policy-map configuration mode.<br>By default, no policy maps are defined. |
| **Step 5** | **class** {*class-map-name* \| **class-default**}<br>**Example:**<br>`Device(config-pmap)# `**`class ipclass1`** | Defines a traffic classification, and enter policy-map class configuration mode.<br>By default, no policy map class-maps are defined.<br>If a traffic class has already been defined by using the **class-map** global configuration command, specify its name for class-map-name in this command. |

| | Command or Action | Purpose |
|---|---|---|
| | | A **class-default** traffic class is predefined and can be added to any policy. It is always placed at the end of a policy map. With an implied **match any** included in the **class-default** class, all packets that have not already matched the other traffic classes will match **class-default**. |
| **Step 6** | **set { cos \| dscp \| ip \| precedence \| qos-group \| wlan user-priority}**<br>**Example:**<br>`Device(config-pmap-c)# set dscp 45` | (Optional) Sets the QoS values. Possible QoS configuration values include:<br>• **cos**—Sets the IEEE 802.1Q/ISL class of service/user priority.<br>• **dscp**—Sets DSCP in IP(v4) and IPv6 packets.<br>• **ip**—Sets IP specific values.<br>• **precedence**—Sets precedence in IP(v4) and IPv6 packet.<br>• **qos-group**—Sets QoS group.<br>• **wlan user-priority**—Sets WLAN user priority.<br>In this example, the **set dscp** command classifies the IP traffic by setting a new DSCP value in the packet. |
| **Step 7** | **police** { *target_bit_rate* \| **cir** \| **rate** }<br>**Example:**<br>`Device(config-pmap-c)# police 100000 conform-action transmit exceed-action drop` | (Optional) Configures the policer:<br>• *target_bit_rate*—Specifies the bit rate per second, enter a value between 8000 and 10000000000.<br>• **cir**—Committed Information Rate.<br>• **rate**—Specifies the police rate PCR for hierarchical policies.<br>In this example, the **police** command adds a policer to the class where any traffic beyond the 100000 set target bit rate is dropped. |
| **Step 8** | **exit**<br>**Example:**<br>`Router(config-pmap-c)# exit` | Returns to policy map configuration mode. |
| **Step 9** | **exit**<br>**Example:**<br>`Router(config-pmap)# exit` | Returns to global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 10** | **interface** [*interface-id*]<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/0` | Specifies the port to attach to the policy map, and enters interface configuration mode.<br>Valid interfaces include physical ports. |
| **Step 11** | **service-policy input** [*policy-map-name*]<br>**Example:**<br>`Device(config-if)# service-policy input flowit` | Specifies the policy-map name, and applies it to an ingress port. Only one policy map per ingress port is supported. |
| **Step 12** | **end**<br>**Example:**<br>`Router(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 13** | **show policy-map** [*policy-map-name* [**class** *class-map-name*]]<br>**Example:**<br>`Router(config)# show policy-map` | (Optional) Verifies your entries. |
| **Step 14** | **copy running-config startup-config**<br>**Example:**<br>`Router(config)# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

### What to do next

If applicable to your QoS configuration, configure classification, policing, and marking of traffic on SVIs by using policy maps.