Layer 2 Configuration Guide, Cisco IOS XE 17 (Cisco ASR 920 Series)

First Published: 2019-11-14

Last Modified: 2022-04-11

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

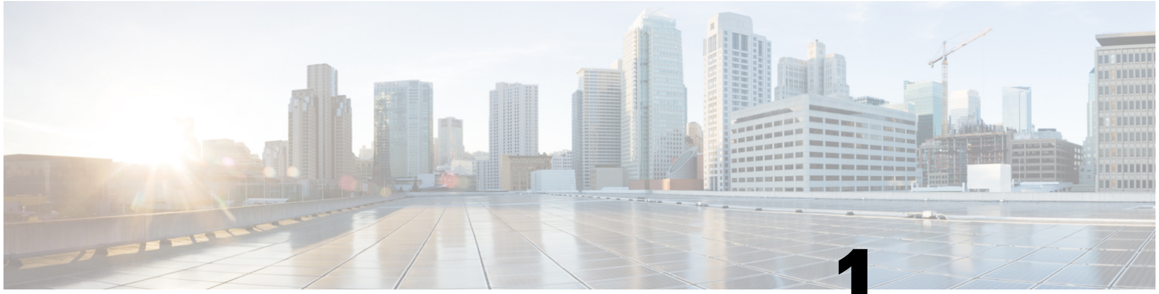
Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

CONTENTS



CHAPTER 1

Feature History

The following table lists the new and modified features supported in the Layer 2 Configuration Guide in Cisco IOS XE 17 releases.

Feature	Description
Cisco IOS XE Cupertino 17.8.1	
EDPL support on interfaces configured with 802.1ad.	This feature allows EDPL functionality to be supported on interfaces that are configured with 802.1ad encapsulation.
Support for Ethernet Data Plane Loopback on Bundle Interface	This feature enables ethernet data plane loopback on bundle interfaces. You can also configure the feature when the router is not physically connected and the port is in down state. This feature is only applicable on internal or terminal loopback in up or down state.
Cisco IOS XE Bengaluru 17.4.1	
Enhanced Ethernet Data Plane Loopback	The Ethernet data plane loopback feature is enhanced to avoid control packets getting dropped. The enhancement supports internal shaper configuration, when terminal ELB session is activated or deactivated to rate the limit the ELB session traffic.

CHAPTER 2

Configuring Switched Port Analyzer

This document describes how to configure local Switched Port Analyzer (SPAN) and remote SPAN (RSPAN) on the router.

Prerequisites for Configuring Local SPAN and RSPAN

Local SPAN

- Use a network analyzer to monitor interfaces.

RSPAN

- Before configuring RSPAN sessions, you must first configure:

  1. Source interface
  2. Destination Bridge Domain over VPLS

Restrictions for Local Span and RSPAN

Local Span

- Local SPAN is only supported on physical ports.
- VLAN filtering is not supported.
- SPAN monitoring of port-channel interfaces or port-channel member-links is *not* supported.

- Combined Egress local SPAN bandwidth supported is 1 GB.
- Local SPAN isn’t supported on logical interfaces such as VLANs or EFPs.
- Up to 14 active local SPAN sessions (ingress and egress) are supported. The router supports up to 14 ingress sessions and up to 12 egress sessions.
- Only one local SPAN destination interface is supported. You *can’t* configure a local SPAN destination interface to receive ingress traffic.
- Outgoing Cisco Discovery Protocol (CDP), Bridge Protocol Data Unit (BPDU), IS-IS, and OSPF packets are not replicated.
- When enabled, local SPAN uses any previously entered configuration.
- When you specify source interfaces and do not specify a traffic direction (**Tx**, **Rx**, or **both**), **both** is used by default.
- Local SPAN destinations never participate in any spanning tree instance. Local SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the local SPAN destination are from the local SPAN source.
- Local SPAN sessions with overlapping sets of local SPAN source interfaces or VLANs are *not* supported.
- Configuring SPAN and netflow on the same interface is not supported. If SPAN and netflow have been mistakenly configured on the same interface, reset the interface. Use the **default interface** command to set the interface back to its default values, and then configure SPAN.

  The following sample shows how to reset the interface:

  ```
  router(config)#default interface GigabitEthernet0/0/0
  router(config)#interface GigabitEthernet0/0/0
  router(config)#ip address 192.168.16.1 255.255.255.0
  router(config)#negotiation auto
  router(config)#cdp enable
  ```

  For the SPAN configuration, see Configuring Sources and Destinations for Local SPAN, on page 10.

RSPAN

- RSPAN VLAN/BD is *not* used for data traffic.
- The maximum number of supported RSPAN sessions are 14.
- Only one source port is supported per RSPAN.
- Only port channel RSPAN is supported.
- Per member link RSPAN is not supported.
- Source ranges (VLAN range or port range) is *not* supported.
- VLAN filtering is not supported.
- If two RSPAN configurations sessions are configured on two RSPAN BDs associated to the same Trunk EFP, the traffic from the first session flows to the second session after it is configured.
- RSPAN destination configuration for Layer2 pseudowire is *not* supported.
- If RSPAN BD is associated with a VPLS pseudowire, the traffic flows through the VPLS pseudowire.

- Do not have RSPAN bridge domain as part of RSPAN source interface.
- RSPAN spans the Rx traffic even when the classifying service instance of the receiving port is in admin down state.
- If RSPAN source and destinations are separated by pseudowire, then the RSPAN details must be updated on both RSPAN source switch and destination switch. The pseudowire should also be dedicated for RSPAN traffic.
- Source and destination ports for a Tx SPAN or RSPAN session should be in the same ASIC. This is applicable to Cisco RSP2 module.

Note Incomplete configuration of RSPAN / LSPAN will result in traffic drop issues.

Understanding Local SPAN and RSPAN

Information About Local SPAN Session and RSPAN Session

Local SPAN Session

A local Switched Port Analyzer (SPAN) session is an association of a destination interface with a set of source interfaces. You can configure local SPAN sessions to monitor all traffic in a specified direction. Local SPAN sessions allow you to monitor traffic on one or more interfaces and to send either ingress traffic, egress traffic, or both to one destination interface.

Local SPAN sessions do not interfere with the normal operation of the switch. You can enable or disable SPAN sessions with command-line interface (CLI) commands. When enabled, a local SPAN session might become active or inactive based on various events or actions, and this would be indicated by a syslog message. The **show monitor session span session number** command displays the operational status of a SPAN session.

A local SPAN session remains inactive after system power-up until the destination interface is operational.

The following configuration guidelines apply when configuring local SPAN on the router:

- When enabled, local SPAN uses any previously entered configuration.
- Use the **no monitor session** *session number* command with no other parameters to clear the local SPAN session number.

Local SPAN Traffic

Network traffic, including multicast, can be monitored using SPAN. Multicast packet monitoring is enabled by default. In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination interface. For example, a bidirectional (both ingress and egress) SPAN session is configured for sources a1 and a2 to a destination interface d1. If a packet enters the switch through a1 and gets switched to a2, both incoming and outgoing packets are sent to destination interface d1; both packets would be the same (unless a Layer-3 rewrite had occurred, in which case the packets would be different).

RSPAN Session

An RSPAN source session is an association of source ports or VLAN across your network with an RSPAN Vlan. The RSPAN VLAN/BD on the router is the destination RSPAN session.

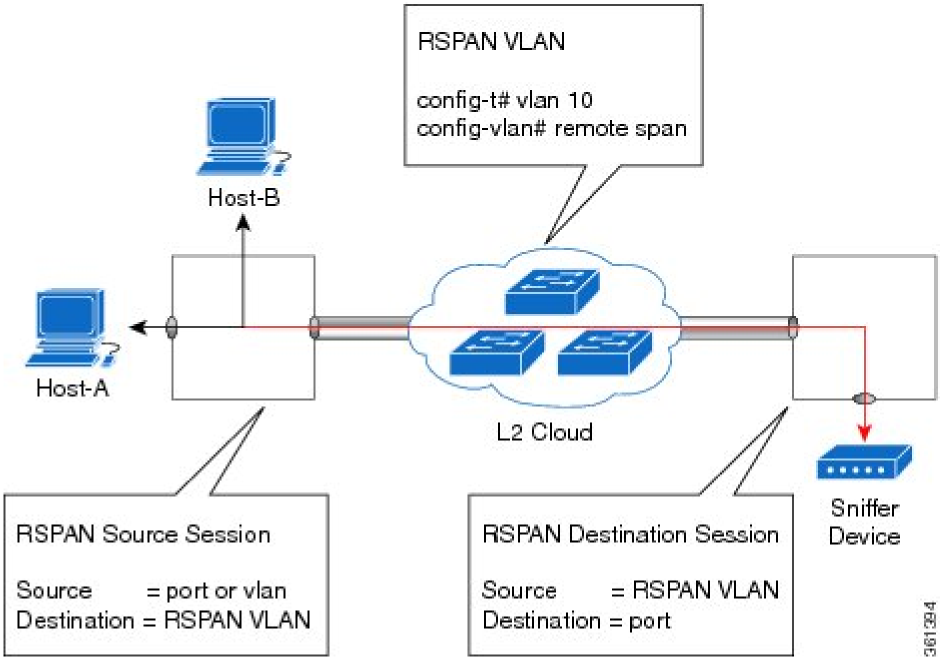
RSPAN Traffic for RSP2 Module

RSPAN supports source ports and source VLANs in the source switch and destination as RSPAN VLAN/BD.

The figure below shows the original traffic from the Host A to Host B via the source ports or VLANs on Host A. The source ports or VLANs of Host A is mirrored to Host B using RSPAN VLAN 10. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating devices. The traffic from the source ports or VLANs are mirrored into the RSPAN VLAN and forwarded over Trunk or the EVC bridge domain (BD) ports carrying the RSPAN VLAN to a destination session monitoring the RSPAN VLAN.

Each RSPAN source must have either ports or VLANs as RSPAN sources. On RSPAN destination, the RSPAN VLAN is monitored and mirrored to the destination physical port connected to the sniffer device.

Figure 1: RSPAN Traffic



RSPAN allows remote monitoring of traffic where the source and destination switches are connected by L2VPN networks

The RSPAN source is either ports or VLANs as in a traditional RSPAN. However, the SPAN source and destination devices are connected through a L2 pseudowire associated with the RSPAN VLAN over an MPLS/IP network. The L2 pseudowire is dedicated for only RSPAN traffic. The mirrored traffic from the source port or VLAN is carried over the pseudowire associated with the RSPAN VLAN towards the destination side. On the destination side, a port belonging to the RSPAN VLAN or EVC BD is connected to sniffer device.

Destination Interface

A destination interface, also called a monitor interface, is a switched interface to which SPAN or RSPAN sends packets for analysis. You can have only one destination interface for SPAN sessions.

An interface configured as a destination interface cannot be configured as a source interface. Specifying a trunk interface as a SPAN or RSPAN destination interface stops trunking on the interface.

Source Interface

A source interface is an interface monitored for network traffic analysis. An interface configured as a destination interface cannot be configured as a source interface.

Traffic Directions

Ingress SPAN (Rx) copies network traffic received by the source interfaces for analysis at the destination interface. Egress SPAN (Tx) copies network traffic transmitted from the source interfaces to the destination interface. Specifying the configuration option (both) copies network traffic received and transmitted by the source interfaces to the destination interface.

The following table lists the supported traffic types for RSPAN.

Table 1: RSPAN over VPLS Traffic for RSP3 module

Source	Ingress Mirror (Rx)	Egress Mirror (Tx)	Both
CFM	Not Supported	Supported	Not Supported
Layer 2	Supported	Supported	Supported
Layer 3	Incoming Ethernet and VLAN header are stripped off and RSPANed over VPLS	Supported	Not Supported
L2VPN	Not Supported	Supported	Not Supported
L3VPN	Not Supported	Supported	Not Supported
L3VPN over BDI	Not Supported	Supported	Not Supported
MPLS	Incoming Ethernet and VLAN header are stripped off and RSPANed over VPLS	Supported	Not Supported
Routed PW	Not Supported	Supported	Not Supported
VPLS	Not supported for bidirectional traffic	Supported	Not Supported

Table 2: RSPAN Traffic

Source	Ingress Mirror (Rx)	Egress Mirror (Tx)	Both
Layer2 or Layer3	Supported	Supported	Supported

Source	Ingress Mirror (Rx)	Egress Mirror (Tx)	Both
VLAN	Supported	Not supported	Not supported
EFP	Not supported	Not supported	Not supported
Pseudowire	Not supported	Not supported	Not supported

The following table lists the supported **rewrite** traffic for RSPAN on the EFP, Trunk with the associated RSPAN Bridge Domains (BD).

Table 3: Rewrite Traffic for RSPAN BD

Rewrite Operations	Source	EFP/Trunk associated with RSPAN BD
no-rewrite	Pop1, Pop2, Push1	Only Pop1

The following tables lists the format of the spanned packets at the destination port for both Ingress and Egress RSPAN. The tables lists the formats of untagged, single, and double tagged source packets for EFPs under source port configured with **rewrite** operations (no-rewrite, pop1, pop2 and push1).

Table 4: Destination Port Ingress and Egress Spanned Traffic for EVC RSPAN BD

	Ingress Traffic	Egress Traffic
(Untagged Traffic) - Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + packet	RSPAN BD tag + packet
pop1 tag	NA	NA
pop2 tag	NA	NA
push1 tag	NA	NA
(Single Traffic)-Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + source-outer-tag + packet	RSPAN BD tag + source-outer-tag + packet
pop1 tag		
pop2 tag		NA
push1 tag		RSPAN BD tag + source-outer-tag + packet
(Double traffic) - Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**

	Ingress Traffic	Egress Traffic
no-rewrite	RSPAN BD tag + source-outer-tag + source-inner-tag + packet	RSPAN BD tag + Source-inner-tag + packet
pop1 tag		
pop2 tag		
push1 tag		

Table 5: Destination Port Ingress and Egress Spanned Traffic for TEFP RSPAN BD

	Ingress Traffic	Egress Traffic
(Untagged traffic)- Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + packet	RSPAN BD tag + packet
pop1 tag	NA	NA
pop2 tag	NA	NA
push1 tag	NA	NA
(Single traffic)-Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + source-outertag + packet	RSPAN BD tag + source-outertag + packet
pop1 tag		
pop2 tag		NA
push1 tag		RSPAN BD tag + source-outertag + packet
(Double traffic) -Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + source-outertag + source-innertag+ packet	RSPAN BD tag + source-outertag + source-innertag + packet
pop1 tag		
pop2 tag		
push1 tag		

Table 6: Destination Port Ingress and Egress Spanned Traffic for RSPAN BD with VPLS Pseudowire (RSP2 module)

	Ingress Traffic	Egress Traffic
(Untagged traffic) - Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**

	Ingress Traffic	**Egress Traffic**
no-rewrite	RSPAN BD tag + packet	RSPAN BD tag + packet
pop1 tag	NA	NA
pop2 tag	NA	NA
push1 tag	NA	NA
(Single traffic)- Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + source-outer-tag + packet	RSPAN BD tag + source-outer-tag + packet
pop1 tag		
pop2 tag	NA	NA
push1 tag	RSPAN BD tag + source-outer-tag + packet	RSPAN BD tag + source-outer-tag + packet
(Double traffic)-Source port rewrite	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**	**RSPAN VLAN (BD) rewrite pop1 tag symmetric**
no-rewrite	RSPAN BD tag + source-outer-tag + source-inner-tag + packet	RSPAN BD tag + source-outer-tag + source-inner-tag + packet
pop1 tag		
pop2 tag		
push1 tag		

Configuring Local SPAN and RSPAN

Configuring Sources and Destinations for Local SPAN

To configure sources and destinations for a SPAN session:

Procedure

Step 1 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 2 **monitor session {*session_number*} type local**

Example:

```
Router(config)# monitor session 1 type local
```

Specifies the local SPAN session number and enters the local monitoring configuration mode.

- *session_number*—Indicates the monitor session. The valid range is 1 through 14.

Step 3 **source interface** *interface_type slot/subslot/port* **[, | - | rx | tx | both]**

Example:

```
Router(config-mon-local)# source interface gigabitethernet 0/2/1 rx
```

Specifies the source interface and the traffic direction:

- *interface_type*—Specifies the Gigabit Ethernet or Ten Gigabit Ethernet interface.
    - *slot/subslot/port*—The location of the interface.
- “,”—List of interfaces
- “–”—Range of interfaces
- rx—Ingress local SPAN
- tx—Egress local SPAN
- both

Step 4 **destination interface** *interface_type slot/subslot/port* **[, | -]**

Example:

```
Router(config-mon-local)# destination interface gigabitethernet 0/2/4
```

Specifies the destination interface that sends both ingress and egress local spanned traffic from source port to the prober or sniffer.

- *interface_type*—Specifies the Gigabit Ethernet or Ten Gigabit Ethernet interface.
    - *slot/subslot/port*—The location of the interface.
- “,”—List of interfaces
- “–”—Range of interfaces

Step 5 **no shutdown**

Example:

```
Router(config-mon-local)# no shutdown
```

Enables the local SPAN session.

Step 6 **End**

Removing Sources or Destinations from a Local SPAN Session

To remove sources or destinations from a local SPAN session, use the following commands beginning in EXEC mode:

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **no monitor session** *session-number*

Example:

```
Router(config)# no monitor session 2
```

Clears existing SPAN configuration for a session.

Configuring RSPAN Source Session

To configure the source for a RSPAN session:

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **monitor session** *RSPAN_source_session_number* **type rspan-source**

Example:

```
Router(config)# monitor session 1
 type rspan-source
```

Configures an RSPAN source session number and enters RSPAN source session configuration mode for the session.

- *RSPAN_source_session_number—*

  Valid sessions are 1 to 14.
- **rspan-source**—Enters the RSPAN source-session configuration mode.

Step 4 **Filter vlan***vlan id*

Example:

```
filter vlan 100
```

Applies the VLAN access map to the VLAN ID; valid values are from 1 to 4094.

Step 5 **source** {*single_interface* slot/subslot/port| *single_vlan* [**rx** | **tx** | **both**]

Example:

```
Router(config-mon-rspan-src)# source interface gigabitethernet 0/2/1 tx
```

Specifies the RSPAN session number, the source interfaces and the traffic direction to be monitored.

- *single_interface—*Specifies the Gigabit Ethernet or Ten Gigabit Ethernet interface.
  - *slot/subslot/port*—The location of the interface.
- *single_vlan*

  —Specifies the single VLAN.
- **both**

  —(Optional) Monitors the received and the transmitted traffic.
- **rx**

  —(Optional) Monitors the received traffic only.
- **tx**—(Optional) Monitors the transmitted traffic only.

Step 6 **destination remote vlan** *rspan_vlan_ID*

Example:

```
Router(config-mon-rspan-src)# destination remote vlan2
```

Associates the RSPAN source session number session number with the RSPAN VLAN.

- *rspan_vlan_ID*—Specifies the Vlan ID.

  Note *rspan_vlan_ID* is the RSPAN BD that is configured under the EFP or port which carries the RSPANd traffic.

Step 7 **no shutdown**

Example:

```
Router(config-mon-rspan-src)# no shutdown
```

Enables RSPAN source.

Step 8 **end**

Example:

```
Router(config-mon-rspan-src)# end
```

Exists the configuration.

Configuring RSPAN Destination Session

To configure the destination for a RSPAN session for remote Vlan:

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **monitor session** *RSPAN_destination_session_number* **type rspan-destination**

Example:

```
Router(config)# monitor session 1 type rspan-destination
```

Configures a RPAN session.

- *RSPAN_destination_session_number*—Valid sessions are 1 to 80.
- **rspan-destination**—Enters the RSPAN destination-session configuration mode.

Step 4 **source remote vlan** *rspan_vlan_ID*

Example:

```
Router(config-mon-rspan-dst)# source remote vlan2
```

Associates the RSPAN destination session number RSPAN VLAN.

- *rspan_vlan_ID*—Specifies the Vlan ID

Step 5 **destination** {*single_interface slot/subslot/port}*

Example:

```
Router(config-mon-rspan-dst)# destination interface gigabitethernet 0/0/1
```

Associates the RSPAN destination session number with the destination port.

- *single_interface* —Specifies the Gigabit Ethernet or Ten Gigabit Ethernet interface.
  - *slot/subslot/port*—The location of the interface.

Step 6 **no shutdown**

Example:

```
Router(config-mon-rspan-dst)# no shutdown
```

Restarts the interface

Step 7 **end**

Example:

```
Router(config-mon-rspan-dst)# end
```

Exists the configuration

Removing Sources or Destinations from a RSPAN Session

To remove source or destination from a RSPAN session, delete and recreate the RSPAN session. The following are the steps:

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **no monitor session** *session number*

Example:

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

Exits monitor session.

Step 4 **end**

Example:

```
Router(config-mon-rspan-src)# end
```

Exits configuration mode.

Sample Configurations

The following sections contain configuration example for SPAN and RSPAN on the router.

Configuration Example: Local SPAN

The following example shows how to configure local SPAN session 8 to monitor bidirectional traffic from source interface Gigabit Ethernet interface to destination:

```
Router(config)# monitor session 8 type local
Router(config)# source interface gigabitethernet 0/0/10
Router(config)# destination interface gigabitethernet 0/0/3
Router(config)# no shut
```

Configuration Example: Removing Sources or Destinations from a Local SPAN Session

This following example shows how to remove a local SPAN session:

```
Router(config)# no monitor session 8
```

Configuration Example: RSPAN Source

The following example shows how RSPAN session 2 to monitor bidirectional traffic from source interface Gigabit Ethernet 0/0/1:

```
Router(config)# monitor session 2 type RSPAN-source
Router(config-mon-RSPAN-src)# source interface gigabitEthernet0/0/1 [tx |rx|both]
Router(config-mon-RSPAN-src)# destination remote VLAN 100
Router(config-mon-RSPAN-src)# no shutdown
Router(config-mon-RSPAN-src)# end
```

The following example shows how RSPAN session 3 to monitor bidirectional traffic from source Vlan 200:

```
Router(config)# monitor session 3 type RSPAN-source
Router(config-mon-RSPAN-src)# filter vlan 100
Router(config-mon-RSPAN-src)# source interface Te0/0/23 rx
Router(config-mon-RSPAN-src)# destination remote VLAN 200
Router(config-mon-RSPAN-src)# no shutdown
Router(config-mon-RSPAN-src)# end
```

Configuration Example: RSPAN Destination

The following example shows how to configure interface Gigabit Ethernet 0/0/1 as the destination for RSPAN session 2:

```
Router(config)# monitor session 2 type RSPAN-destination
Router(config-mon-RSPAN-dst)# source remote VLAN 100
Router(config-mon-RSPAN-dst)# destination interface gigabitEthernet 0/0/1
Router(config-mon-RSPAN-dst)# end
```

Verifying Local SPAN and RSPAN

Use the **show monitor session** command to view the sessions configured.

- The following example shows the Local SPAN source session with Tx as source:

```
Router# show monitor session 8
Session 8
---------
Type : Local Session
Status : Admin Enabled
Source Ports :
TX Only : Gi0/0/10
Destination Ports : Gi0/0/3
MTU : 1464
Dest RSPAN VLAN : 100
```

- The following example shows the RSPAN source session with Gigabit Ethernet interface 0/0/1 as source:

```
Router# show monitor session 2
Session 2
---------
Type                    : Remote Source Session
Status                  : Admin Enabled
Source Ports            :
    Both                : Gi0/0/1
MTU                     : 1464
```

- The following example shows the RSPAN source session with Vlan 20 as source:

```
Router# show monitor session 3
Session 3
---------
Type                    : Remote Source Session
Status                  : Admin Enabled
Source VLANs            :
    RX Only             : 20
MTU                     : 1464
```

- The following example shows the RSPAN destination session with Gigabit Ethernet interface 0/0/1 as destination:

```
Router# show monitor session 2
Session 2
---------
Type                     : Remote Destination Session
Status                   : Admin Enabled
Destination Ports        : Gi0/0/1
MTU                      : 1464
Source RSPAN VLAN : 100
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 3

Layer 2 Access Control Lists on EVCs

The ability to filter packets in a modular and scalable way is important for both network security and network management. Access Control Lists (ACLs) provide the capability to filter packets at a fine granularity. In Metro Ethernet networks, ACLs are directly applied on Ethernet virtual circuits (EVCs).

Layer 2 Access Control Lists on EVCs is a security feature that allows packet filtering based on MAC addresses. This module describes how to implement ACLs on EVCs.

Prerequisites for Layer 2 Access Control Lists on EVCs

- Knowledge of how service instances must be configured.
- Knowledge of extended MAC ACLs and how they must be configured.

Restrictions for Layer 2 Access Control Lists on EVCs

- You can enable a packet capture on the host, based on Layer 2 packet header per EFP (for example, **dst-mac**, **src-mac** and CoS field). Create a **pcap** of the captured packet on host machine.
- A maximum of 16512 access control entries (ACEs) are allowed for a given ACL, with the limitation that it does not exceed the maximum tcam entries.
- Only 256 different or unique Layer 2 ACLs can be configured on a line card. (More than 256 ACLs can be configured on a router and it depends on the number of TCAM that is free for programming these ACLs.)
- L2 ACL is supported over port channel with Normal EFPs.
- Egress L2 ACL on EVC is *not* supported.
- L2 ACLs are *not* supported on Trunk EFP.

- L2 ACL counters are *not* supported.
- Layer2 ACL can be applied on layer 2 frame without IPv4 or IPv6 header as layer 2 ACL does not support filter on IPv4 or IPv6 traffic.
- Layer 2 ACLs function inbound only. The Layer 2 ACLs are *not* supported at physical interface level.
- Current Layer 2 ACLs provide Layer 3 filtering options in permit and deny rules. Options that are not relevant to service instances are ignored.

Information About Layer 2 Access Control Lists on EVCs

EVCs

An Ethernet virtual circuit (EVC) as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. It is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer. An EVC contains the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a specified port.

Service instances are configured under a port channel. The traffic carried by the service instance is load balanced across member links. Service instances under a port channel are grouped and each group is associated with one member link. Ingress traffic for a single EVC can arrive on any member of the bundle. All egress traffic for a service instance uses only one of the member links. Load balancing is achieved by grouping service instances and assigning them to a member link.

Ethernet virtual connection services (EVCS) uses the EVCs and service instances to provide Layer 2 switched Ethernet services. EVC status can be used by a customer edge (CE) device either to find an alternative path to the service provider network or in some cases, to fall back to a backup path over Ethernet or over another alternative service such as ATM.

For information about the Metro Ethernet Forum standards, see the Standards table in the “Additional References” section.

Relationship Between ACLs and Ethernet Infrastructure

The following points capture the relationship between ACLs and Ethernet Infrastructure (EI):

- ACLs can be directly applied on an EVC using the command-line interface (CLI). An ACL is applied to a service instance, which is the instantiation of an EVC on a given port.
- One ACL can be applied to more than one service instance at any time.
- One service instance can have one ACL at most applied to it at any time. If a Layer 2 ACL is applied to a service instance that already has a Layer 2 ACL, the new one replaces the old one.
- Only named ACLs can be applied to service instances. The command syntax ACLs is retained; the **mac access-list extended** command is used to create an ACL.
- The **show ethernet service instance id** *id* **interface** *type number* detail **show ethernet service instance** command can be used to provide details about ACLs on service instances.

How to Configure Layer 2 Access Control Lists on EVCs

Creating a Layer 2 ACL

Perform this task to create a Layer 2 ACL with a single ACE.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mac access-list extended** *name*

Example:

```
Device(config)# mac access-list extended test-12-acl
```

Defines an extended MAC ACL and enters mac access list control configuration mode.

Step 4 **permit** {{*src-mac mask* | **any**} {*dest-mac mask* | **any**} [*protocol* [**vlan** *vlan*] [*cos value*]]}

Example:

```
Device(config-ext-macl)# permit 00aa.00bb.00cc 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. Creates an ACE for the ACL.

Applying a Layer 2 ACL to a Service Instance

Perform this task to apply a Layer 2 ACL to a service instance. Note that packet filtering takes place only after the ACL has been created and applied to the service instance.

Before you begin

Before applying an ACL to a service instance, you must create it using the **mac access-list extended command. See the "Creating a Layer 2 ACL" section.**

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet 1/0/0
```

Specifies the type and location of the interface to configure, where:

- *type* --Specifies the type of the interface.
- *number* --Specifies the location of the interface.

Step 4 **service instance** *id* ethernet

Example:

```
Device(config-if)# service instance 100 ethernet
```

Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.

Step 5 **encapsulation dot1q** *vlan-id*

Example:

```
Device(config-if-srv)# encapsulation dot1q 100
```

Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.

Step 6 **mac access-group** *access-list-name* in

Example:

```
Device(config-if-srv)# mac access-group test-12-acl in
```

Applies a MAC ACL to control incoming traffic on the interface.

Step 7 **bridge -domain** *bridge-id* in

Example:

```
Device(config-if-srv)# bridge-domain 100
```

Configure the bridge domain ID.

Configuring a Layer 2 ACL with ACEs on a Service Instance

Perform this task to configure the same ACL with three ACEs and stop all other traffic on a service instance.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **mac access-list extended** *name*

Example:

```
Device(config)# mac access list extended test-12-acl
```

Defines an extended MAC ACL and enters mac access control list configuration mode.

Step 4 **permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**}

Example:

```
Device(config-ext-macl)# permit 00aa.bbcc.ddea 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.

Step 5 **permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**}

Example:

```
Device(config-ext-macl)# permit 00aa.bbcc.ddeb 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.

Step 6 **permit** {*src-mac mask* | **any**} {*dest-mac mask*} | **any**}

Example:

```
Device(config-ext-macl)# permit 00aa.bbcc.ddec 0.0.0 any
```

Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.

Step 7 **deny any any**

Example:

```
Device(config-ext-macl)# deny any any
```

Prevents forwarding of Layer 2 traffic except for the allowed ACEs.

Step 8 **exit**

Example:

```
Device(config-ext-macl)# exit
```

Exits the current command mode and returns to global configuration mode.

Step 9 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet 1/0/0
```

Specifies the interface.

Step 10 **service instance** *id* **ethernet**

Example:

```
Device(config-if)# service instance 200 ethernet
```

Configures an Ethernet service instance on an interface and enters service instance configuration mode.

Step 11 **encapsulation dot1q** *vlan-id*

Example:

```
Device(config-if-srv)# encapsulation dot1q 100
```

Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.

Step 12 **mac access-group** *access-list-name* **in**

Example:

```
Device(config-if-srv)# mac access-group test-12-acl in
```

Applies a MAC ACL to control incoming traffic on the interface.

Verifying the Presence of a Layer 2 ACL on a Service Instance

Perform this task to verify that a Layer 2 ACL is present on an EVC. This verification task can be used after an ACL has been configured to confirm its presence.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 configure terminal

Example:

```
Device# show ethernet service instance id 100 interface gigabitethernet 3/0/1 detail
```

Enters global configuration mode.

Step 3 **show ethernet service instance id** *id* **interface** *type* *number* detail

Example:

```
Device# show ethernet service instance id 100 interface gigabitethernet 3/0/1 detail
```

Displays detailed information about Ethernet customer service instances.

Configuration Examples for Layer 2 Access Control Lists on EVCs

Example Applying a Layer 2 ACL to a Service Instance

The following example shows how to apply a Layer 2 ACL called mac-20-acl to a service instance. The ACL has five permitted ACEs and all other traffic is not allowed.

```
enable
configure terminal
 mac access-list extended mac-20-acl
 permit 00aa.bbcc.adec 0.0.0 any

 permit 00aa.bbcc.bdec 0.0.0 any

 permit 00aa.bbcc.cdec 0.0.0 any

 permit 00aa.bbcc.edec 0.0.0 any

 permit 00aa.bbcc.fdec 0.0.0 any

 deny any any
 exit
interface gigabitethernet 10/0/0
 service instance 100 ethernet
 encapsulation dot1q 100
 mac access-group mac-20-acl in
```

Example Applying a Layer 2 ACL to Three Service Instances on the Same Interface

The following example shows how to apply a Layer 2 ACL called mac-07-acl to three service instances on the same interface:

```
enable
configure terminal
mac access-list extended mac-07-acl

permit 00aa.bbcc.adec 0.0.0 any

permit 00aa.bbcc.bdec 0.0.0 any

permit 00aa.bbcc.cdec 0.0.0 any

deny any any
exit
interface gigabitethernet 10/0/0
service instance 100 ethernet
encapsulation dot1q 100
mac access-group mac-07-acl in
service instance 101 ethernet
encapsulation dot1q 101
mac access-group mac-07-acl in
service instance 102 ethernet
encapsulation dot1q 102
mac access-group mac-07-acl in
```

Example Creating a Layer 2 ACL with ACEs

The following example shows how to create a Layer 2 ACL called mac-11-acl with two permitted ACEs:

```
enable
```

```
configure terminal
mac access-list extended mac-11-acl
permit 00aa.00bb.00cc 1a11.0101.11c1 any
permit 00aa.00bb.00cc 1a11.0101.11c2 any
```

Example Displaying the Details of a Layer 2 ACL on a Service Instance

The following sample output displays the details of a Layer 2 ACL called test-acl on a service instance.

```
Device# show ethernet service instance id 100 interface ethernet0/0 detail
Service Instance ID: 100
L2 ACL (inbound): test-acl
Associated Interface: Ethernet0/0
Associated EVC: test
L2protocol drop
CEVlans:
Interface Dot1q Tunnel Ethertype: 0x8100
State: Up
L2 ACL permit count: 10255
L2 ACL deny count: 53
```

```
Device# show ethernet service instance id 100 interface gig3/0/1 detail
Service Instance ID: 100
L2 ACL (inbound): test-acl
Associated Interface: Gig3/0/1
Associated EVC: test
L2protocol drop
CEVlans:
Interface Dot1q Tunnel Ethertype: 0x8100
State: Up
L2 ACL permit count: 10255
L2 ACL deny count: 53
```

The table below describes the significant fields in the output.

Table 7: show ethernet service instance Field Descriptions

Field	Description
Service Instance ID	Displays the service instance ID.
L2 ACL (inbound):	Displays the ACL name.
Associated Interface:	Displays the interface details of the service instance.
Associated EVC:	Displays the EVC with which the service instance is associated.
CEVlans:	Displays details of the associated VLAN ID.
State:	Displays whether the service instance is in an up or down state.
L2 ACL permit count:	Displays the number of packet frames allowed to pass on the service instance by the ACL.
L2 ACL deny count	Displays the number of packet frames not permitted to pass on the service instance by the ACL.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Standards and RFCs

Standard/RFC	Title
MEF 6.1	*Metro Ethernet Services Definitions Phase 2 (PDF 6/08)*
MEF 10.1	*Ethernet Services Attributes Phase 2 (PDF 10/06)*

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

CHAPTER 4

Configuring Ethernet Dataplane Loopback

Ethernet data plane loopback provides a means for remotely testing the throughput of an Ethernet port.

Prerequisites for Ethernet Data Plane Loopback

- Ethernet loopback sessions are supported only of EFPs (service instances, Ethernet flow points, EVCs).
- Dot1q tags are not configured for default and untagged EFPs.
- Ethernet loopback sessions are supported on dot1q or QinQ or untagged and default EFPs.
- Internal loopback sessions configured must be within the 1 GB reserved bandwidth.
- Internal loopback can be launched even when the physical interface port state is down.

Restrictions for Ethernet Data Plane Loopback

- If the facility loopback is active on either Nile 0 or Nile 1, then only the Ingress QoS policy works on this facility. Egress QoS for facility loopback does not work due to the platform restriction.
- Facility loopback behavior on Gibraltar: Ingress and egress QoS policies on the EFP/TEFP gets bypassed. There is no support to bypass Ingress/Egress Port level policies as it works as configured.
- Data plane loopback on routed port infrastructure is *not* supported.

- Etype, src-mac, or llc-oui based loopback traffic filtering is *not* supported.
- Port-level QoS is not bypassed. The egress port shaper cannot be bypassed.
- Port shaper on the ingress port in both external and internal loopback cannot be bypassed.
- Ethernet loopback is not supported on a range of dot1q tags.
- Default EFP loopback is *not* supported in the shutdown state.
- Loopback sessions cannot be initiated on a port that is configured with SPAN or RSPAN.
- During Internal loopback, MAC swap is not supported for multicast or broadcast traffic.
- Only one Ethernet loopback (terminal or facility) session can be active on an EFP at any instance.
- Egress SPAN on the port and internal loopback on an EFP on the same port cannot be configured at the same time.
- Egress ACL is not supported on the EFP.
- A maximum number of 20 facility loopback sessions can be created per system, provided 16 sessions are with dot1q and 4 sessions are with dot1q and destination MAC address. This scale reduces if SPAN or RSPAN is configured.
- A maximum number of 12 terminal loopback sessions can be created per system, provided 8 sessions are with dot1q and 4 sessions are with dot1q and destination MAC address. This scale reduces if RSPAN or SADT is configured.
- Internal Ethernet Data Plane Loopback session can also be launched when the interface or port is in down state.
- We recommended to avoid performing any dynamic changes to the interface state when the Ethernet Data Plane Loopback (ELB) is configured on a port that is in the down state. There is a behavior change when interface is moved from up to down state, as internal ELB session will not be stopped or removed.
- Ethernet data plane Loopback is not supported with the XConnect service when the physical interface port state is down.
- Ethernet data plane Loopback will be affected on STP enabled interface.
- Dynamic addition of rewrite ingress tags with default EFP is not supported.
- Dynamic changes at EFP and interface level are not supported when Ethernet Data Plane Loopback is active.
- dot1q tag inclusion in the configuration for default and untagged EFP disables the Ethernet Data Plane Loopback.
- When loopback is configure for a default EFP on the interface, then all the traffic (ingressing) in this interface gets looped back.
- BFD flaps on enabling internal loopback and traffic looped back with line rate as both the traffic passes through the HPCT queue.
- If traffic is more than 650Mbps and if the packet size is less than a frame size of 64, then BFD and OSPF flaps are expected.

Information on Ethernet Data Plane Loopback

The Ethernet data plane loopback feature provides a means for remotely testing the throughput of an Ethernet port. You can verify the maximum rate of frame transmission with no frame loss. This feature allows for bidirectional or unidirectional throughput measurement, and on-demand/out-of-service (intrusive) operation during service turn-up. This feature supports two types of Ethernet loopback. RSP3 supports the following types of loopback from Cisco IOS XE Everest 16.5.1 release.

- Facility loopback (external)—Traffic loopback occurs at the Ingress interface. Traffic does not flow into the router for loopback.
- Terminal loopback (internal)—Traffic loopback occurs at the Egress interface. Traffic loopback occurs after the traffic flows into the router to the other interface.

QoS Support for Ethernet Data Plane Loopback

- Ingress QoS is bypassed in external loopback on service instances.
- Internal loopback sequence is as follows:
  - Ingress QoS
  - Egress QoS (egress port) (both, shaper and policer are supported).
  - Ingress QoS on ingress port and egress QoS on egress port (both, shaper and policer are supported) on the RSP3 module.
  - Ingress QoS on egress port and egress QoS on ingress port on the RSP3 module.
- All port-level and EFP-level QoS is applicable for internal Ethernet data plane loopback.
- For external Ethernet data plane loopback:
  - All port-level and EFP-level QoS is bypassed except for shaper.
  - Port-level shaper cannot be bypassed.

How to Configure Ethernet Data Plane Loopback on Physical Interfaces

Enabling Ethernet Data Plane Loopback on Physical Interfaces

Table 8: Feature History

Feature Name	Release Information	Feature Description
EDPL support on dot1ad	Cisco IOS XE Cupertino 17.8.1	This features enables configuration of Ethernet Data Plane Loopback on interfaces configured with 802.1ad encapsulation. This helps measure the interface throughput handing encapsualted traffic.

```
enable
configure terminal
interface gigabitethernet 0/2/1
service instance 1 ethernet
encapsulation dot1ad 101 dot1q 100
bridge-domain 120
ethernet loopback permit external
end
```

Note ELB is supported using a MAC filter for UP-MEP session. If you are starting ELB without the MAC filter, the UP-MEP session will go DOWN.

Starting an Ethernet Data Plane Loopback Session on Physical Interfaces

Note To start a loopback for untagged and default EFPs, dot1q and second-dot1q are not needed. Dot1q is *not* applicable to start a loopback session on the RSP3 module.

Note By default the session would be running for 300 seconds unless you explicitly specify and automatically stops after the session time expiry.

```
enable
configure terminal
ethernet loopback start local interface gigabitEthernet 0/4/1 service instance 10 external
 dot1q 10 cos 1 destination mac-address 0000.0000.0001 timeout none
end
This is an intrusive loopback and the packets matched with the service will not be able
```

```
to pass through.
Continue? (yes/[no]): yes
```

Dot1q and COS-based filtering is not supported on the RSP3 module.

```
enable
configure terminal
ethernet loopback start local interface gigabitEthernet 0/4/1 service instance 10 external
 destination mac-address 0000.0000.0001 timeout none
end
```

Configuration Examples

Example: Configuring External Loopback on Physical Interfaces

This example shows how to configure external (facility) loopback.

```
Router(config)# interface gigabitEthernet 0/4/1
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation  dot1q 120
Router(config-if-srv)# bridge-domain 120
Router(config-if-srv)# ethernet loopback permit external
```

This example shows external (facility) loopback on the Gigabit Ethernet 0/4/1 interface:

```
interface GigabitEthernet0/4/1
 no ip address
 negotiation auto
 service instance 10 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
  ethernet loopback permit external ===? For facility loopback
 !
end
```

This example below shows how to start external (facility) loopback on the router. A warning message is displayed. Type **yes** to continue.

```
Router# ethernet loopback start local interface gigabitEthernet 0/4/1 service instance 10
external dot1q 10 cos 1
 destination mac-address 0000.0000.0001 timeout none

This is an intrusive loopback and the packets matched with the service will not  be able
to pass through.
Continue? (yes/[no]): yes
```

Note Dot1q and COS-based filtering is not supported on the RSP3 module.

Example: Configuring Terminal Loopback on Physical Interfaces

This example shows how to configure internal (terminal) loopback.

```
Router(config)# interface gigabitEthernet 0/0/0
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 120
Router(config-if-srv)# bridge-domain 120
Router(config-if-srv)# ethernet loopback permit internal
```

This example shows internal (terminal) loopback on Gigabit Ethernet 0/0/0 interface:

```
interface TenGigabitEthernet0/0/0
 no ip address
service instance 10 ethernet
  encapsulation dot1q 10
  rewrite ingress tag pop 1 symmetric
  bridge-domain 10
  ethernet loopback permit internal
 !
end
```

Verifying Ethernet Data Plane Loopback

Example: Verifying Ethernet Dataplane Loopback on Physical Interfaces

Use the **show ethernet loopback** {**active** | **permitted**} [**interface** *interface numbe*r] command.

- The following example displays the loopback capabilities per interface. The output shows internal (terminal) loopback has been permitted on Ten Gigabit Ethernet 0/0/0 interface and external (facility) loopback has been permitted on Gigabit Ethernet 0/4/1 interface.

```
Router# show ethernet loopback permitted

--------------------------------------------------------------------------------
Interface                               SrvcInst Direction
Dot1q/Dot1ad(s)                         Second-Dot1q(s)
--------------------------------------------------------------------------------
Te0/0/0                                 10                             Internal
10
Gi0/4/1                                 10                             External
10
```

- This example shows all active sessions on the router.

```
Router# show ethernet loopback active

============================================================
Loopback Session ID        : 1
Interface                  : GigabitEthernet0/4/1
Service Instance           :10
Direction                  : External
Time out(sec)              : none
Status                     : on
Start time                 : 10:31:09.539 IST Mon Aug 26 2013
Time left                  : N/A
Dot1q/Dot1ad(s)            : 10
Second-dot1q(s)            :
Source Mac Address         : Any
Destination Mac Address    : 0000.0000.0001
Ether Type                 : Any
Class of service           : 1
Llc-oui                    : Any
```

```
Total Active Session(s)    : 1
Total Internal Session(s)  : 0
Total External Session(s)  : 1
```

- This example shows how to stop the sessions on the router.

```
Router# ethernet loopback stop local interface GigabitEthernet
0/4/1 id 1
```

Information on Enhanced Ethernet Data Plane Loopback

Table 9: Feature History

Feature Name	Release	Description
Enhanced Ethernet Data Plane Loopback	Cisco IOS XE Bengaluru 17.4.1	The Ethernet data plane loopback feature is enhanced to avoid control packets getting dropped. The enhancement supports internal shaper configuration, when terminal ELB session is activated or deactivated to rate the limit the ELB session traffic. The enhancement is applicable only on internal loopback.

The Ethernet data plane loopback feature is enhanced to avoid control packets getting dropped, besides the ELB traffic drop. If terminal ELB is configured on 1 GB interface, then other priority traffic like BFD is dropped due to congestion.

The enhancement supports internal shaper configuration, when terminal ELB session is activated or deactivated to limit the rate for ELB session traffic.

Restrictions

- After starting EDPL, even if you remove ELB_SHAPE from EFP, then EDPL continues to run.
- Do not modify the ELB_SHAPE directly.
  - Use the platform command to make any changes in ELB_SHAPE PM.
  - If you directly modify the ELB_SHAPE PM, then the shaper is applied based on the updated values that are defined under ELB_SHAPE.
- If you remove ELB_SHAPE PM from the global config accidentally, then you must remove the **platform edpl_internal_shaper xxx** platform command and reapply.
- For any change to the EDPL parameters, we recommended you to perform the following:
  1. Deactivate or Stop the EDPL session.
  2. Do the required configuration changes.
  3. Activate or Start the EDPL session.

Do not dynamically change or delete the EDPL shaper or the interface bandwidth.

- In an xconnect scenario, when the interface goes down, the xconnect goes down, but EDPL remains active. Hence, you must stop the EDPL session and start back the EDPL session, after the xconnect is UP.

Configuring Ethernet Data Plane Loopback Session

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **configure EDPL shaper**

Example:

```
Device# platform edpl_internal_shaper  600000000
```

Enables EDPL shaper.

Step 4 **interface GigabitEthernet**

Example:

```
Device(config)# interface GigabitEthernet 0/0/9
Device(config-if)# service instance 1401 ethernet
Device(config-if-srv)# encapsulation dot1q 1401
Device(config-if-srv)# rewrite ingress tag pop 1 symmetric
Device(config-if-srv)# ethernet loopback permit internal
Device(config-if-srv)# end
```

Configures internal loopback on the interface.

Step 5 **ethernet loopback start local**

Example:

```
Device# ethernet loopback start local interface GigabitEthernet 0/0/9 service instance 1401
 internal dot1q 1401 timeout none
```

Starts EDPL loopback session.

Example: Configuring Ethernet Loopback Active

The following example shows the activated and running EDPL session:

```
Router# show ethernet loopback active

Load for five secs: 5%/1%; one minute: 6%; five minutes: 6%
Time source is NTP, 18:18:23.680 IST Wed Aug 26 2020

============================================================
Loopback Session ID     : 1
Interface               : GigabitEthernet0/0/9
Service Instance        : 1401
Direction               : Internal
Time out(sec)           : none
Status                  : on
Start time              : 18:17:58.360 IST Wed Aug 26 2020
Time left               : N/A
Dot1q/Dot1ad(s)         : 1401
Second-dot1q(s)         :
Source Mac Address      : Any
Destination Mac Address : Any
Ether Type              : Any
Class of service        : Any
Llc-oui                 : Any

Total Active Session(s): 1
Total Internal Session(s): 1
Total External Session(s): 0
```

To deactivate an EDPL session.

```
Router# ethernet loopback stop local interface gigabitEthernet 0/0/9 id 1

003846: Aug 26 18:40:56.528 IST: %E_DLB-6-DATAPLANE_LOOPBACK_STOP: Ethernet Dataplane
Loopback Stop on interface GigabitEthernet0/0/9 service instance 1401 with session id 1
```

Example: Configuring Ethernet Data Plane Loopback Session

The following example shows the global shaper configuration:

Note If the interface bandwidth is less than the shaper value, then EDPL does not get started. You can refer to the syslogs.

```
Interface bandwidth 400Mb, shaper value 600Mb

platform edpl_internal_shaper  600000000

interface GigabitEthernet0/0/9
 bandwidth 400000
```

```
service instance 1401 ethernet
  encapsulation dot1q 1401
  rewrite ingress tag pop 1 symmetric
  xconnect 192.211.92.3 1401 encapsulation mpls
  ethernet loopback permit internal
 !

Router# ethernet loopback start local interface gigabitEthernet 0/0/9 service instance 1401
 internal dot1q 1401 timeout none

This is an intrusive loopback and the packets matched with the service will not be able to
 pass through. Continue? (yes/[no]): yes
QoS Configuration failed !!! Shaper value is greater than interface speed

 Failed to config EDPL shaper, EDPL not activated !!!
```

Use Cases or Deployment Scenarios

ELB is Supported with MAC Filter for UP-MEP Session

In the following scenario, ELB is supported using a MAC filter for UP-MEP session. If you starting ELB with out MAC filter, the UP-MEP session will go DOWN.

```
enable
configure terminal
service instance 800 ethernet 800
encapsulation dot1q 800
service-policy input <NAME>
xconnect 10.0.0.2 880 encapsulation mpls
cfm mep domain <NAME> mpid 200
cos 7
ethernet loopback permit external
ethernet loopback permit internal

Router#ethernet loopback start local interface gi0/0/0 service instance 800 internal dot1q
 800 destination mac-address f078.1685.313f timeout none

This is an intrusive loopback and the packets matched with the service will not  be able
to pass through. Continue? (yes/[no]): yes


Router#show ethernet cfm maintenance-points remote
--------------------------------------------------------------------------------
MPID  Domain Name                                 MacAddress          IfSt  PtSt
 Lvl  Domain ID                                   Ingress
 RDI  MA Name                                     Type Id             SrvcInst
      EVC Name                                                        Age
      Local MEP Info
--------------------------------------------------------------------------------
220   CCI                                         f078.1685.313f      Up    Up
 0    CCI                                         Gi0/0/0:(10.0.0.2, 880)
 -    800                                         XCON N/A            800
      800                                                             0s
      MPID: 200 Domain: CCI MA: 800

Total Remote MEPs: 1
```

Support for Ethernet Data Plane Loopback on Bundle Interface

Table 10: Feature History

Feature Name	Release Information	Feature Description
Support for Ethernet Data Plane Loopback on Bundle Interface	Cisco IOS XE Cupertino 17.8.1	This feature enables ethernet data plane loopback on bundle interfaces. You can also configure the feature when the router is not physically connected and the port is in down state. This feature is only applicable on internal or terminal loopback in up or down state.

Bundle interface or a link bundle is a group of one or more ports that are aggregated together and treated as a single link. This allows you to group multiple point-to-point links together into one logical link and provide higher bidirectional bandwidth, redundancy, and load balancing between two routers. A virtual interface is assigned to the bundled link. The component links can be dynamically added and deleted from the virtual interface. The virtual interface is treated as a single interface on which you can configure an IP address and other software features used by the link bundle. Packets sent to the link bundle are forwarded to one of the links in the bundle.

Bundle interfaces increase bandwidth availability, because traffic is forwarded over all available members of the bundle. Therefore, traffic can if one of the links within a bundle fails. can without interrupting packet flow. The ethernet dataplane loopback feature configured on bundle interfaces provides a methodology to verify the maximum rate of frame transmission with no frame loss.

Prior to Cisco IOS XE Cupertino 17.8.1, you could only configure ethernet data plane loopback on the physical interfaces.

Starting with Cisco IOS XE Cupertino 17.8.1, you can also configure ethernet data plane loopback feature on the bundle interfaces. But, you can only configure internal or terminal loopback in up or down state.

This feature is only supported on Cisco RSP2 module.

Scenario: Support for Ethernet Dataplane Loopback on Link Down Port

Consider a scenario when you need to configure ethernet dataplane loopback feature before the router is physically connected. Thus, you need to configure the feature when the port link is down. Starting with Cisco IOS XE Cupertino 17.8.1, you can configure the terminal or internal ethernet dataplane loopback feature even when the router is not physically connected and the link is down. But, as this feature is not supported on external or facility loopback, you cannot configure external loopback feature when the port link is down.

A port channel bundles individual interfaces into a group to provide increased bandwidth and redundancy. Previously, the internal ethernet dataplane loopback was not supported on the port channel interface. Starting with Cisco IOS XE Cupertino 17.8.1, you can now configure internal loopback on the port channel interface even when the interface is down.

Restrictions for Ethernet Dataplane Loopback on Bundle Interface

- The internal ethernet dataplane loopback feature is only available with service instance for port channel interface. It is not available when you configure MPLS or IP or Layer 3 on port channel interface.
- At least one member link must be added to the port channel interface for ethernet dataplane loopback.
- External ethernet loopback session on port channel interface is *not* supported.
- This feature will only function for traffic flow on first member of the port channel.
- You cannot configure the feature when the bundle members are in suspended state.
- The maximum traffic performance of terminal loopback is 1GBPS.

Configure Ethernet Dataplane Loopback Start Session on Bundle Interface

1. **Activate Loopback on the EFP**

   To activate terminal loopback on the EFP:

   ```
   interface Po1
    no ip address
   service instance 10 ethernet
     encapsulation dot1q 10
     rewrite ingress tag pop 1 symmetric
     bridge-domain 10
     ethernet loopback permit internal  === For Terminal Loopback
    !
   ```

2. **Start Loopback Session**

   To start a terminal loopback session on bundle interface:

   ```
   R11#ethernet loopback start local in po1 ser ins 2 inte dot1q 2 destination mac-address
    3333.0001.0003 tim non

   This is an intrusive loopback and the packets matched with the service will not  be able
    to pass through. Continue? (yes/[no]): yes
   ```

Configure Ethernet Dataplane Loopback Stop Session on Bundle Interface

1. **Stop Loopback Session**

   To stop a terminal loopback session on a bundle interface:

   ```
   R2#ethernet loopback stop local interface po1 id 1
   ```

2. **Deactivate Loopback Session on the EFP**

   To stop the terminal loopback session in the EFP:

   ```
   interface Po1
    no ip address
    service instance 10 ethernet
     encapsulation dot1q 10
     rewrite ingress tag pop 1 symmetric
     bridge-domain 10
     no ethernet loopback permit internal
   ```

Verification of Ethernet Dataplane Loopback Configuration on Bundle Interface

Use the **show ethernet loopback active** command to display all active sessions on the router.

```
R11#show ethernet loopback active
============================================================
Loopback Session ID      : 1
Interface                : Port-channel1
Service Instance         : 2
Direction                : Internal
Time out(sec)            : none
Status                   : on
Start time               : 10:35:16.940 IST Fri Dec 17 2021
Time left                : N/A
Dot1q/Dot1ad(s)          : 2
Second-dot1q(s)          :
Source Mac Address       : Any
Destination Mac Address  : 3333.0001.0003
Ether Type               : Any
Class of service         : Any
Llc-oui                  : Any


Total Active Session(s): 1
Total Internal Session(s): 1
Total External Session(s): 0
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Ethernet Dataplane Loopback

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for Ethernet Dataplane Loopback

Feature Name	Releases	Feature Information
Ethernet Dataplane Loopback	Cisco IOS XE Release 3.14.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D, ASR-920-10SZ-PD, ASR-920-24SZ-IM, ASR-920-24SZ-M, ASR-920-24TZ-M) .

CHAPTER 5

Static MAC Address Support on Service Instances

The Multicast and Unicast static MAC address support on Service Instances feature supports configuration of a static MAC address on a pseudoport. Use of a static MAC address for Broadband Network Gateway (BNG) upstream traffic enables traffic forwarding while conserving MAC table resources and limiting the traffic flood by creating multicast groups.

Prerequsites for Static MAC Address Support on Service Instances

- Knowledge of both port and bridge domain limitations.
- Knowledge of service instances.

Restrictions for Static MAC Address Support on Service Instances

- Multicast static MAC addresses are not allowed in MAC address security configurations.
- Static MAC addresses are programmed only on switch processors (both active and standby).
- Static MAC configuration is *not* allowed at secure service instance.
- Static MAC addresses are programmed only on switch processors (both active and standby).

- The Static MAC address on Pseudowires is *not* supported on the Cisco ASR 900 Series Routers.
- Static MAC address configuration is *not* supported on Trunk EFP.

Information about Static MAC Address Support on Service Instances

Static MAC address configuration on service instances eliminates the need for MAC address learning, which is required for traffic forwarding. In the upstream direction, without MAC address learning, MAC address table resources can be conserved and network resources optimized.

When a bridge domain ID is either changed or deleted for a service instance, all static MAC addresses are removed.

When a service instance is deleted, all static MAC addresses on that pseudoport are removed.

Configuring a Static MAC Address on a Service Instance

Perform this task to manually configure a static MAC address on a service instance.

Procedure

Step 1 **enable**

Example:

```
Router> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Router# configure terminal
```

Enters global configuration mode.

Step 3 **interface** *type number*

Example:

```
Router(config)# interface Ethernet 1/0GigabitEthernet
 0/2/1
```

Configures an interface type and enters interface configuration mode.

Step 4 **service instance** *id* **ethernet** [*evc-id*]

Example:

```
Router(config-if)# service instance 1 ethernet
```

Configures an Ethernet service instance on an interface and enters service instance configuration mode.

Step 5 **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**]

Example:

```
Router(config-if-srv)# encapsulation dot1q 100
```

Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.

Step 6 **bridge-domain** *bridge-id* [**split-horizon**[**group** *group-id*]]

Example:

```
Router(config-if-srv)# bridge-domain 100
```

Binds a service instance to a bridge domain instance.

Step 7 **mac static address** *mac-addr* [**auto-learn**] [**disable-snooping**]

Example:

```
Router(config-if-srv)# mac static address 0000.bbbb.cccc
```

Configures a static MAC address.

Step 8 **exit**

Example:

```
Router(config-if-srv)# exit
```

Returns the CLI to privileged EXEC mode.

Example for Configuring a Static MAC Address on a Service Instance

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet 0/2/1
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac static address 0000.bbbb.cccc
Router(config-if-srv)# exit
```

Verifying Configured Static MAC Addresses on a Service Instance

Use one or more of the following commands to verify the configured static MAC address on a service instance:

- **show bridge-domain**
- **show mac address-table**

Example: Verifying Configured Static MAC Addresses on a Service Instance

show bridge-domain

The sample output for the **show bridge-domain** command:

```
Router# show bridge-domain 10 mac static address

Bridge-Domain ID : 10
Static MAC count : System : 1, bridge-domain : 1

Port                                     Address          Action
Gi0/3/7 ServInst 10                      aaa1.123c.bc32
```

show mac address-table

The sample output for the **show mac address-table** command:

```
Router# show mac address-table bdomain 10

   Nile Mac Address Entries

   BD    mac addr        type     ports

------------------------------------------------------------------------------------------

   10    aaa1.123c.bc32  STATIC   Gi0/3/7.Efp10
```

Additional References

Related Documents

Related Topic	Document Title
Configuration guide	*Cisco IOS Carrier Ethernet Configuration Guide*, Cisco IOS XE Release (ASR 903)
Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*

Related Topic	Document Title
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

CHAPTER 6

MAC Limiting

This document describes how to configure MAC limiting.

Information About Global MAC Address Limiting on Bridge Domain

Table 12: Feature History

Feature Name	Release Information	Description
Mac Address Limiting Per Bridge Domain	Cisco IOS XE Cupertino 17.8.1	This feature restricts the number of MAC addresses that the router learns in a bridge-domain on an EFP or trunk EFP to a specified number. Use the feature to enable warning and limit actions when a violation occurs.

MAC address limiting per bridge-domain restricts the number of MAC addresses that the router learns in a bridge-domain on an EFP or trunk EFP to a specified number.

Note For the RSP1 and RSP2 modules, the local connect feature is not supported on the Cisco router. However, to simulate a local connect scenario, configure the connecting EFPs on the same bridge domain and disable the mac-learning on the bridge domain by setting the MAC limit to 0. Use the **mac-address-table limit bdomain** *num* **maximum** *0* **action limit** command to disable mac-learning on bridge-domain.

When the total number of MAC addresses (dynamic MAC addresses) in a bridge-domain exceeds the maximum number, then the router takes a violation action. The router either restricts further learning on bridge-domain by itself with a syslog or just intimate the user through a syslog to take further action.

You can enable the following actions when violation occurs:

- Warning—The violation is logged as a syslog message and no further action is taken. There is one syslog message received, when the MAC count exceeds the configured limit (exceed notification) and no more syslog messages are received for the bridge-domain (bdomain) unless the violation is no longer valid (drop notification). When you select the warning action, the further learning of new MAC addresses and forwarding of traffic continue to happen irrespective of violation.
- Limit—When the Limit option is selected as an action for violation, the MAC learning on the bdomain is disabled when violation occurs. No new MAC addresses are learnt on the bdomain until the recovery mechanism gets started. Even though new MAC addresses are not learned but frames are still flooded in the system. If user needs to stop flooding, then a sub action flood can also be used along with limit action.

Note The threshold value must be 80% of the maximum value configured for the recovery mechanism.

- Flood—The flood sub action allows the user to disable unknown unicast flooding on a given bdomain. This flood sub action is initiated only when the limit action is configured and violation has occurred. Unknown unicast flooding is disabled only for the interval necessary to limit the entries. Using this option, improves the performance and the flooding is re-enabled when the total number of MAC entries are dropped below the threshold value.
- Shutdown—When the shutdown action is selected, a syslog message is generated and the particular bdomain on which violation occurred is disabled. Hence, all the learning and forwarding of traffic are stopped on the bdomain. The bdomain remain in such state until the feature is explicitly disabled through CLI.

Note **Warning** is the default action when no action is configured.

Note The functionality of automatic error recovery is *not* supported on the Cisco ASR 900 RSP2 module.

For the limit and warning actions, the recovery mechanism is initiated when the total MAC limit count drops to equal or below a threshold value. The threshold value is dependent on the maximum limit configured on bridge domain (the threshold value is 80% of the limit value). The recovery mechanism reverts the action taken during violation. For example, if the MAC address learning is disabled as a violation action, then it will be re-enabled.

If no maximum value or action option in specified through the **mac address-table limit bdomain id maximum num action** command, then the default action (warning) and a default maximum value of 500 is configured.

Note For a MAC limit of 0 with the action limit, limit flood, the violation action occurs when the user configures it irrespective of MAC address learning on the bridge domain. The recovery mechanism is to disable the feature through the **no mac address-table limit bdomain id** command.

Restrictions and Usage Guidelines for the RSP1 and RSP2 Modules

MAC limiting is supported on the following interface types:

- You can apply MAC limiting only to bridge-domains.
- MAC limiting is supported for dynamic MAC addresses.

Restrictions for MAC Limiting for RSP3 Module

- Bridge domain MAC limit and EFP MAC Security are not supported together on a bridge domain.
- The change in split horizon group configuration is not supported on the bridge domain if the MAC limit is already configured on that domain.
- A maximum number of four unique MAC limit values can be configured at any time. Many bridge domains can use the same values but it cannot be shared with a bridge domain interface. If the bridge domain interface is added to the existing bridge domain MAC limit configuration, then the configuration should be removed and added again.
- On a Trunk EFP, if the violation is noticed on atleast one of the bridge domains, then the violation action applies to the whole Trunk EFP. If one bridge domain has the action limit, the limit flood or the shutdown action exceeds, then the whole Trunk EFP's MAC learning is disabled.
- The allowed MAC limit range is from 0 through 0xFFFD.
- The MAC limit on the bridge domain interface needs to be configured to a value higher than the actual maximum limit value that is expected. This is because an internal static MAC is added if the bridge domain interface has an IP configured or the corresponding bridge domain is a part of L2VPN. This will be taken into account for MAC limit.
- The action warning is applied based on the software learning and a delay of approximately 1 minute is observed while generating syslog on a normal bridge domain.
- The delay in the drop notification in based on the software again and the delay is approximately 1 minute for the syslog generation.
- In case of MAC limit 0, static MACs are allowed to be added even after the limit exceeds, only if the bridge domain is UP.

Configuring MAC Limiting

Procedure

Step 1 **configure terminal**

Enter global configuration mode.

Step 2 **mac-address-table limit** *bdomain id maximum num action {warning | limit | shutdown} [flood]*

Sets the specific limit and any optional actions to be imposed at the bridge-domain level.

The default **maximum** value is 500.

Step 3 **end**

Return to privileged EXEC mode.

Step 4 **show mac-address-table limit bdomain** *bdomain id*

Displays the information about the MAC-address table.

Step 5 **copy running-config startup-config**

(Optional) Save your entries in the configuration file.

Example of Enabling Per-Bridge-Domain MAC Limiting

This example shows how to enable per-bridge-domain MAC limiting.

```
Router# enable
Router# configure terminal
Router(config)# mac-address-table limit bdomain 10 maximum 100 action limit flood
Router(config)# end
```

Verifying the MAC Limiting on Bridge Domain

Use the **show mac address-table limit** command to verify the information related to configured MAC limit per bridge domain.

This example shows how to display the information related to configured MAC limit per bridge domain.

```
Router#show mac address-table limit bdomain 10
  bdomain       action      flood       maximum     Total entries     Current state
-------------+----------+------------+------------+---------------+---------------
   10           limit      Disable        100           0             Within Limit
```