



Configuring Traffic Mirroring

This module describes the configuration of the traffic mirroring feature. Traffic mirroring is sometimes called port mirroring, or switched port analyzer (SPAN).

Feature History for Traffic Mirroring

Release 7.3.1	SPAN to File feature was introduced.
Release 7.2.12	Local SPAN feature was introduced.
Release 7.0.14	Support for the following features was introduced in ERSPAN: <ul style="list-style-type: none">• Configuration of IP DSCP.• Tunnel IP.• Ability to source ranges of interfaces and SVIs.• Sequence bit is set in the GRE header and the value of sequence number is always 0 for ERSPAN packets.• ERSPAN and Security ACL should be separate. Support for File Mirroring was introduced.
Release 7.0.11	This feature was introduced.

- [Introduction to Traffic Mirroring, on page 2](#)
- [Restrictions for Traffic Mirroring, on page 11](#)
- [Configuring Traffic Mirroring, on page 12](#)
- [Attaching the Configurable Source Interface, on page 15](#)
- [Introduction to ERSPAN Rate Limit, on page 17](#)
- [Introduction to File Mirroring, on page 19](#)
- [Introduction to Local SPAN, on page 21](#)
- [Traffic Mirroring with DSCP, on page 25](#)
- [SPAN to File, on page 28](#)
- [Traffic Mirroring Configuration Examples, on page 34](#)
- [Troubleshooting Traffic Mirroring, on page 36](#)

Introduction to Traffic Mirroring

Traffic mirroring, which is sometimes called port mirroring, or Switched Port Analyzer (SPAN) is a Cisco proprietary feature. Traffic mirroring enables you to monitor Layer 3 network traffic passing in, or out of, a set of Ethernet interfaces. You can then pass this traffic to a network analyzer for analysis.

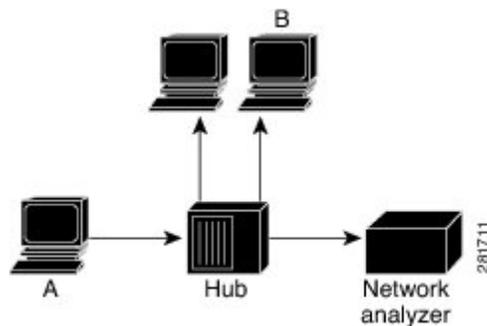
Traffic mirroring copies traffic from one or more Layer 3 interfaces or sub-interfaces. Traffic mirroring then sends the copied traffic to one or more destinations for analysis by a network analyzer or other monitoring device. Traffic mirroring does not affect the switching of traffic on the source interfaces or sub-interfaces. It allows the system to send mirrored traffic to a destination interface or sub-interface.

Traffic mirroring is introduced on switches because of a fundamental difference between switches and hubs. When a hub receives a packet on one port, the hub sends out a copy of that packet from all ports except from the one at which the hub received the packet. In case of switches, after a switch boots, it starts to build up a Layer 2 forwarding table on the basis of the source MAC address of the different packets that the switch receives. After the system builds this forwarding table, the switch forwards traffic that is destined for a MAC address directly to the corresponding port.

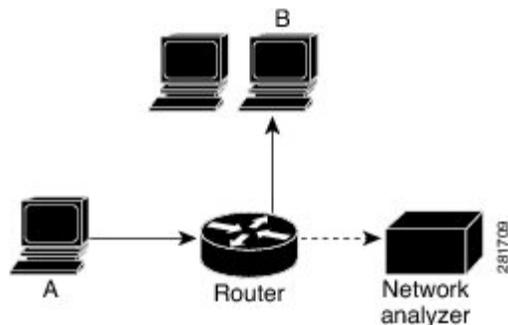
Layer 2 SPAN is not supported on the router.

For example, if you want to capture Ethernet traffic that is sent by host A to host B, and both are connected to a hub, attach a traffic analyzer to this hub. All other ports see the traffic between hosts A and B.

Figure 1: Traffic Mirroring Operation on a Hub



On a switch or router, after the system learns host B MAC address, the system forwards the unicast traffic from A to B to the B port. Therefore, the traffic analyzer does not see this traffic.



In this configuration, the traffic analyzer captures only traffic that is flooded to all ports, such as:

- Broadcast traffic

- Multicast traffic with CGMP or Internet Group Management Protocol (IGMP) snooping disabled
- Unknown unicast traffic on a switch

An extra feature is necessary that artificially copies unicast packets that host A sends. This extra feature is traffic mirroring. When traffic mirroring is enabled, the traffic analyzer is attached to a port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port. The other sections of this document describe how you can fine tune this feature.

Implementing Traffic Mirroring on the Cisco 8000 Series Routers

ERSPAN

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Partial Packet Capture Ability for ERSPAN (RX)	Release 7.5.3	With this feature, you can perform partial packet capture in the RX direction. Earlier, the ability for entire packet capture was available, now you can choose entire or partial packet capture in the RX direction. Here, partial packet capture is also known as truncation.
ERSPAN over MPLS Traffic	Release 7.5.3	With this release, the router allows you to mirror MPLS traffic and set up the GRE tunnel with the next hop over a labeled path. This feature helps you to remote-monitor the traffic on traffic analyzers.
Higher Payload Analysis with Eight ERSPAN Sessions	Release 7.3.2	With this release, Cisco 8000 Series routers support eight ERSPAN sessions. This functionality helps you analyze higher payloads in real time across Layer 3 domains on your network.
ERSPAN over GRE IPv6	Release 7.3.2	With this release, the router allows you to mirror IPv4 or IPv6 traffic with ERSPAN over GRE IPv6 sessions to monitor traffic on remote traffic analyzers. In earlier releases, ERSPAN traffic monitoring was possible only on IPv4 networks

Encapsulated Remote Switched Port Analyzer (ERSPAN) mirrors traffic on one or more source ports and delivers the mirrored traffic to destination port on another switch or management server. ERSPAN enables network operators to troubleshoot issues in the network in real-time using automated tools that auto-configures ERSPAN parameters on the network devices to send specific flows to management servers for in-depth analysis.

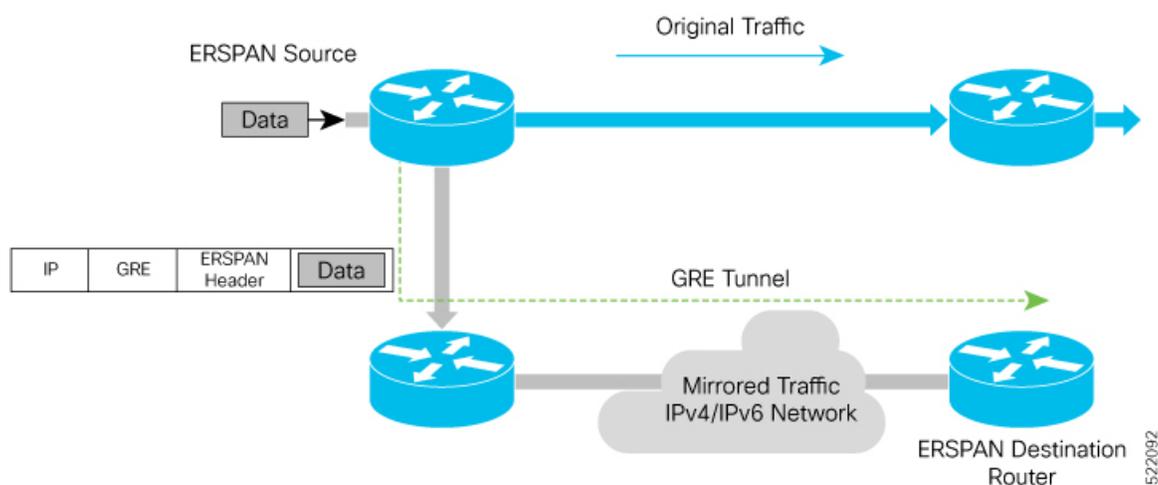
ERSPAN transports mirrored traffic over an IP network. The traffic is encapsulated at the source router and is transferred across the network.

From Cisco IOS XR Software Release 7.5.3 onwards, the packet truncation feature is supported over remote GRE tunnels. You can now get the flexibility to truncate packets and mirror the traffic.

Starting with Cisco IOS XR Software Release 7.0.14, sequence bit is set in the GRE header and the value of sequence number is always 0 for ERSPAN packets.

Starting with Cisco IOS XR Software Release 7.5.3, the sequence number bit will always be set to one and the sequence number field (4 bytes), will always be set to zero.

Figure 2: ERSPAN over GRE



Supported Capabilities

The following capabilities are supported:

- The source interfaces are layer 3 interfaces, such as physical, and bundle interfaces or subinterface.
- The routers mirror IPv4 and IPv6 traffic.
- ERSPAN with GRE IPv4 or IPv6 has tunnel destinations.
- ERSPAN supports only RX direction.
- ERSPAN over GRE IPv4 and IPv6 supports SPAN ACL.
- Supports MPLS traffic mirroring and GRE tunnel configuration with the next hop over a labeled path.
- Each monitor session allows only one destination interface.
- ACL permit or deny entries with capture action are part of mirroring features.
- The next hop interface must be a main interface. It can be a Physical or Bundle interface.

- Supports full packet capture.
- In ERSPAN over GRE IPv6, the **HopLimit** and **TrafficClass** fields in outer IPv6 header are editable under the tunnel configuration.
- The maximum SPAN sessions supported in the Cisco 8000 Router are as follows:.

SPAN Type	7.3.1 and Prior Releases	7.3.2 and Later Releases
ERSPAN (GRE IPv4, GRE IPv6, or GRE IPv4 + GRE IPv6)	4	8
Local SPAN	4	4
SPAN to File	4	4
Combined SPAN (GRE IPv4 + GRE IPv6 + Local SPAN + SPAN to File)	4	8

Supported Capabilities for ERSPAN Packet Truncation support

The following are the capabilities and requirements:

- Ability to enable the new ERSPAN GREv4 and GREv6 truncation configuration per device.
- Truncation configuration should be on the monitor sessions. Packets received from all sources will only be truncated when you configure the truncation on a monitor session.
- By default, the whole packet will be mirrored without the **mirror first <number>** (truncation size) configuration.
- If the monitor session truncation size is less than the configured-truncation size (343 bytes), then whole packet is mirrored.

If the monitor session truncation size exceeds 343 bytes, the configuration is accepted. However, only 343 bytes truncation size is programmed.

An `ios-msg` is displayed to warn the user.

Example: ERSPAN only support 343 bytes truncation size. `monitor-session` with `session_id <id>` will be set to 343 bytes only.

Restrictions

The following are the ERSPAN and SPAN ACL restrictions:

- The ERSPAN mirror packet is received with a TTL minus 1.
The mirror packet is not identical to the incoming packet and TTL minus 1 is the expected value in the ERSPAN packet.
- The router mirrors only unicast traffic.
However, from Cisco IOS XR Software Release 7.5.3 onwards, the router can mirror multicast traffic.
- Remove and re-apply monitor-sessions on all interfaces after modifying the access control list (ACL).

- GRE tunnel is only dedicated to ERSPAN mirrored packets.
- Only ERSPAN TYPE II header is supported. The value of the index field is always 0. The value of the session-ID field is an internal number that is used by the data path to distinguish between sessions.
- Traffic accounting of the ERSPAN mirrored packets is not supported.



Note You can view the SPAN packet count per session, using the [show monitor-session status internal](#) command.

- ERSPAN decapsulation is unsupported.
- In Cisco IOS XR releases 7.5.2 and earlier, ERSPAN over GREv4 and ERSPAN over GREv6 are not supported when MPLS LDP configuration are present on the router.
However, from Cisco IOS XR Software Release 7.5.3 onwards, the ERSPAN will be functional regardless of any configuration related to MPLS or LDP present on the router.
- MPLS packet mirroring is supported only from Cisco IOS XR Software Release 7.5.3 onwards.
- Due to data path limitation, the source IPv6 addresses of the outer IPv6 header of the ERSPAN packet have only higher 64 bits as valid. The lower 64-bits value is changed to zero. The destination GREv6 IPv6 address should contain all the 128 bits.

Restrictions for ERSPAN Packet Truncation support

- From Cisco IOS XR Software Release 7.5.3 onwards, you can truncate the packet size only to a fixed value of 343 bytes.

Traffic Mirroring Terminology

- Ingress traffic—Traffic that enters the switch.
- Egress traffic—Traffic that leaves the switch.
- Source port—A port that the system monitors with the use of traffic mirroring. It is also called a monitored port.
- Destination port—A port that monitors source ports, usually where a network analyzer is connected. It is also called a monitoring port.
- Monitor session—A designation for a collection of traffic mirroring configurations consisting of a single destination and, potentially, many source interfaces.

Characteristics of the Source Port

A source port, also called a monitored port, is a switched or routed port that you monitor for network traffic analysis. In a single local or remote traffic mirroring session, you can monitor source port traffic, such as received (Rx) for ingress traffic. Your router can support any number of source ports (up to a maximum number of 800).

A source port has these characteristics:

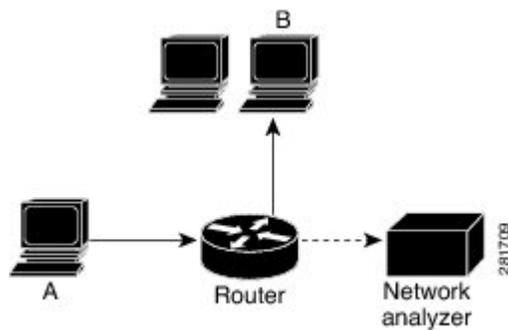
- It can be any port type, such as Bundle Interface, sub-interface, 100-Gigabit Ethernet, or 400-Gigabit Ethernet.



Note Bridge group virtual interfaces (BVI) are not supported.

- Each source port can be monitored in only one traffic mirroring session.
- It cannot be a destination port.
- Each source port can be configured with a direction (ingress) to monitor. For bundles, the monitored direction applies to all physical ports in the group.

Figure 3: Network Analysis on a Router with Traffic Mirroring



In the figure above, the network analyzer is attached to a port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port.

Characteristics of the Monitor Session

A monitor session is a collection of traffic mirroring configurations consisting of a single destination and, potentially, many source interfaces. For any given monitor session, the traffic from the source interfaces (called *source ports*) is sent to the monitoring port or destination port. If there is more than one source port in a monitoring session, the traffic from the several mirrored traffic streams is combined at the destination port. The result is that the traffic that comes out of the destination port is a combination of the traffic from one or more source ports.

Monitor sessions have these characteristics:

- A single monitor session can have only one destination port.
- A single destination port can belong to only one monitor session.



Note The destination of ERSPAN monitoring session is a GRE IPv4 or IPv6 tunnel.

Supported Traffic Mirroring Types

The system supports the following traffic mirroring types:

- ACL-based traffic mirroring. The system mirrors traffic that is based on the configuration of the global interface ACL.
- Layer 3 traffic mirroring is supported. The system can mirror Layer 3 source ports.

ACL-Based Traffic Mirroring

You can mirror traffic that is based on the definition of a global interface access list (ACL). When you are mirroring Layer 3 traffic, the ACL is configured using the **ipv4 access-list** or **ipv6 access-list** command with the **capture** keyword. The **permit** and **deny** commands determine the behavior of regular traffic. The **capture** keyword designates that the packet is to be mirrored to the destination port.

Starting with Cisco IOS XR Software Release 7.0.14, configuration of ERSPAN and security ACL will be separate. Neither of these will have an impact or dependency on the other, but both can be applied simultaneously.

ERSPAN over GRE IPv6

The ERSPAN over GRE IPv6 feature enables mirroring IPv4 or IPv6 traffic in your network. The router encapsulates the traffic adding an ERSPAN header inside the GRE IPv6 packet. The GRE header of the ERSPAN encapsulated packets have the sequence number set to 0. The router sends the replicated traffic packet to be monitored to the destination through the GRE IPv6 channel to achieve traffic mirroring. The mirrored traffic is sent to remote traffic analyzer for monitoring purposes. For the traffic mirroring to work, the ERSPAN GRE IPv6 tunnel next-hop must have ARP or neighbor resolved. We recommend using the `cef proactive-arp-nd enable` command to configure missing adjacency information for the next hop.

```
Router# configure
Router(config)# cef proactive-arp-nd enable
Router(config)# commit
```

Configuring ERSPAN over GRE IPv6

1. Enable GRE IPv6 tunnel configuration.

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#interface tunnel-ipl
RP/0/RP0/CPU0:router(config-if)#tunnel mode gre ipv6
RP/0/RP0/CPU0:router(config-if)#tunnel source 2001:DB8:1::1
RP/0/RP0/CPU0:router(config-if)#tunnel destination 2001:DB8:2::1
RP/0/RP0/CPU0:router(config-if)#no shut
RP/0/RP0/CPU0:router(config)#commit
```

2. Enable ERSPAN session.

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#destination interface tunnel-ipl
RP/0/RP0/CPU0:router(config-mon)#commit
RP/0/RP0/CPU0:router(config-mon)#end
```

3. Configure ERSPAN session under port to be monitored.

```
RP/0/RP0/CPU0:router(config)#interface HundredGigE0/1/0/14
RP/0/RP0/CPU0:router(config-if)#monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)#exit
RP/0/RP0/CPU0:router(config-if)#exit
RP/0/RP0/CPU0:router(config)#interface Bundle-Ether1
RP/0/RP0/CPU0:router(config-if)#monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)#exit
RP/0/RP0/CPU0:router(config-if)#exit
```



```
RP/0/RP0/CPU0:router(config)#interface HundredGigE0/1/0/15.100
RP/0/RP0/CPU0:router(config-subif)#monitor-session mon1 ethernet direction rx-only
```

Verification

Use the `show monitor-session status` command to verify the configuration of the ERSPAN over GRE IPv6 feature.

```
P/0/RP0/CPU0:router#show monitor-session mon1 status
Monitor-session mon1
Destination interface tunnel-ip1
```

```
=====
Source Interface          Dir      Status
-----
Hu0/1/0/14                Rx      Operational
Hu0/1/0/15.100           Rx      Operational
BE1                       Rx      Operational
BE1.1                     Rx      Operational
```

```
RP/0/RP0/CPU0:R1-SF-D#show monitor-session erspan3 status internal
```

```
Thu Jul 15 06:00:14.720 UTC
```

```
Information from SPAN Manager and MA on all nodes:
```

```
Monitor-session erspan3 (ID 0x00000007) (Ethernet)
```

```
SPAN Mgr: Destination interface tunnel-ip372 (0x0f00049c)
```

```
Last error: Success
```

```
Tunnel data:
```

```
Mode: GREoIPv6
```

```
Source IP: 77:3:1::79
```

```
Dest IP: 95::90
```

```
VRF:
```

```
ToS: 100
```

```
TTL: 200
```

```
DFbit: Not set
```

```
0/3/CPU0: Destination interface tunnel-ip372 (0x0f00049c)
```

```
Tunnel data:
```

```
Mode: GREoIPv6
```

```
Source IP: 77:3:1::79
```

```
Dest IP: 95::90
```

```
VRF:
```

```
ToS: 100
```

```
TTL: 200
```

```
DFbit: Not set
```

```
0/RP0/CPU0: Destination interface tunnel-ip372 (0x0f00049c)
```

```
Tunnel data:
```

```
Mode: GREoIPv6
```

```
Source IP: 77:3:1::79
```

```
Dest IP: 95::90
```

```
VRF:
```

```
ToS: 100
```

```
TTL: 200
```

```
DFbit: Not set
```

```
Information from SPAN EA on all nodes:
```

```
Monitor-session 0x00000007 (Ethernet)
```

```
0/3/CPU0: Name 'erspan3', destination interface tunnel-ip372 (0x0f00049c)
```

```
Platform, 0/3/CPU0:
```

```
Monitor Session ID: 7
```

```
Monitor Session Packets: 2427313444
```

```
Monitor Session Bytes: 480591627492
```

Configuring Partial Packet Capture Ability for ERSPAN (RX)

To configure partial traffic mirroring, use the **mirror first** command in monitor session configuration mode.

`Mirror first <number>`: Configures the size of truncation packets for an ERSPAN session

Use the following command to create a ERSPAN monitor session for mirroring the packets:

```
monitor-session <name> [ethernet]
destination interface tunnel-ip <number>
mirror first <number>
    traffic-class <traffic-class>
```

Configuration Example

Use the following command to create a ERSPAN monitor session for mirroring packets to Tunnel-IP 30 with truncation enabled:

```
monitor-session mon1 ethernet
    destination interface tunnel-ip 30
    mirror first 343
!
```

Attach the session to the interfaces using the following configuration:

```
interface <>
    monitor-session session-name ethernet direction rx-only|tx-only|both | acl [acl_name]
```

Running Configuration

```
interface tunnel-ip30
    tunnel mode gre ipv4
    tunnel source 2.2.2.2
    tunnel destination 200.0.0.2
!

interface HundredGigE0/0/0/12
    ipv4 address 12.0.0.2 255.255.255.0
    monitor-session mon1 ethernet direction rx-only
!
```

Verification

The **show monitor-session status internal** displays the size of the programmed truncation.

Example:

```
Router#show monitor-session mon1 status internal
Fri Apr 12 18:50:45.006 UTC
Information from SPAN Manager and MA on all nodes:
Packet truncation size: 343B
Monitor-session mon1 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface Tunnel-IP 20 (0x0f000250)
        Last error: Success

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/RP0/CPU0: Name 'mon1', destination interface Tunnel-IP 20 (0x0f000250)
Platform, 0/RP0/CPU0:

Monitor Session Packets: 142462

Monitor Session Bytes: 7653237
```

Restrictions for Traffic Mirroring

The system supports the following forms of traffic mirroring:

- Mirroring traffic to a GRE IPv4 or IPv6 tunnel (also known as Encapsulated Remote Switched Port Analyzer [ER-SPAN] in Cisco IOS Software). The system allows 8 monitor sessions for ERSPAN, 4 monitor sessions for Local SPAN, and 4 monitor sessions for SPAN to File. The total number of monitor sessions for all SPAN features is 8.
- The system does not support traffic mirroring counters per interface.
- The system does not support bundle member interfaces as sources for mirroring sessions.
- ERSPAN tunnel statistics is not supported.

The following general restrictions apply to traffic mirroring using ACLs:

- Configure ACLs on the source interface to avoid default mirroring of traffic. If a Bundle interface is a source interface, configure the ACLs on the bundle interface (not bundle members).

The following restrictions apply to ERSPAN ACL:

- ERSPAN next-hop must have ARP resolved.
 - Any other traffic or protocol triggers ARP.
- ERSPAN decapsulation is not supported.
- ERSPAN does not work if the GRE next hop is reachable over subinterface. For ERSPAN to work, the next hop must be reachable over the main interface.
- When you use the same ACEs defined in both the IPv4 and IPv6 ACLs, the router doesn't perform ERSPAN mirroring for the ACLs with the lowest priority set as 2 ms.
- However, from Cisco IOS XR Software Release 7.5.3 onwards, GRE next hop can be resolved over subinterface or the main interface.

Modifying ERSPAN monitor-session configuration

When you modify the ERSPAN monitor-session configuration, the **show configuration** and **show configuration commit changes** command outputs differ. Specifically, the **show configuration commit changes** command output displays some extraneous ACL commands deleted and added back. This modified output doesn't impact your configuration or affect performance. This issue is fixed in Cisco IOS XR Release 7.5.1.

The following example highlights the extraneous ACL commands under the **show configuration commit changes** command output.

```
Router(config)#interface HundredGigE0/1/0/0
Router(config-if)#no monitor-session ERSPANtun2005
Router(config-if)#monitor-session ERSPANtun2 ethernet direction rx-only port-level
Router(config-if-mon)#acl
Router(config-if-mon)#acl ipv4 erspan-filter
Router(config-if-mon)#acl ipv6 erspan-filter-ipv6
Router(config-if-mon)#
Router(config-if-mon)#show configuration
```

```

Building configuration...
!! interface HundredGigE0/1/0/0
  monitor-session ERSPANtun2 ethernet direction rx-only port-level
  acl
    acl ipv4 erspan-filter
    acl ipv6 erspan-filter-ipv6
  !
!
end

Router(config-if-mon) #commit
Router(config-if-mon) #end
Router#sh configuration commit changes las 1
Building configuration...
!!
interface HundredGigE0/1/0/0
  no monitor-session ERSPANtun2005 ethernet direction rx-only port-level
  monitor-session ERSPANtun2 ethernet direction rx-only port-level
  no acl
  acl
    no acl ipv4 erspan-filter
    acl ipv4 erspan-filter
    no acl ipv6 erspan-filter-ipv6
    acl ipv6 erspan-filter-ipv6
  !
!
end

```

Configuring Traffic Mirroring

These tasks describe how to configure traffic mirroring:

Configuring ACLs for Traffic Mirroring

This section describes the configuration for creating ACLs for traffic mirroring. You must configure the global interface ACLs by using one of the following commands with the **capture** keyword:

- **ipv4 access-list**
- **ipv6 access-list**



Note Starting with Cisco IOS XR Software Release 7.0.14, ACL feature will provide a support of separate ACL configuration for SPAN.

Configuration

- **Security ACL**

Use the following configuration to configure ACLs for traffic mirroring.

```

/* Create an IPv4 ACL (TM-ACL) for traffic mirroring */
Router(config)# ipv4 access-list TM-ACL
Router(config-ipv4-acl)# 10 permit udp 10.10.10.0 0.0.0.255 eq 10 any capture
Router(config-ipv4-acl)# 20 permit udp 10.10.10.0 0.0.0.255 eq 20 any
Router(config-ipv4-acl)# exit

```

```

Router(config)# commit

/* Apply the traffic monitoring to SPAN source interface */
Router(config)# interface HundredGigE0/0/0/12
Router(config-if)# monitor-session mon1 ethernet direction rx-only port-level acl
Router(config-if)# ipv4 access-group TM-ACL ingress
!

```

Use the following configuration as an example to deny data forwarding for an ACE entry, but still mirror the traffic:

```

ipv4 access-list acl1
10 deny ipv4 any 2.1.0.0/16 capture
20 permit ipv4 any any
!

```

If `acl1` is attached to the interface as shown below:

```

RP/0/RP0/CPU0(config-if)# ipv4 access-group acl1 ingress

```

Data Traffic to 2.1.0.0/16 is dropped. Mirroring happens only if `icmp-off` keyword is added to the ACE as shown below. If this keyword is not added, mirroring does not take place. Furthermore, the `icmp-off` workaround is applicable only to security ACL.

```

ipv4 access-list acl1
10 deny ipv4 any 2.1.0.0/16 capture icmp-off
20 permit ipv4 any any
!

```

• SPAN ACL

- SPAN ACL does not support User Defined Fields (UDF).
- Deny action in SPAN ACL is ignored, and no packet drops from SPAN ACL. Deny ACEs will be internally converted to permit ACEs. Packets will also be mirrored.
- There is no implicit deny-all entry in SPAN ACL.
- IPV6 ACL is required for mirroring IPV6 packet, if IPV4 ACL is configured, and vice versa. This follows the same structure as Security ACL with IPv4 and IPv6 mirror options.

The maximum scale for SPAN ACL ID for fixed and centralized chassis is 3/slice pair and 9/NP. For distributed chassis, the maximum scale for SPAN ACL ID is 6/NP.

Use the following configuration to enable traffic mirroring with ACLs.

```

/* Create a SPAN IPv4 ACL (v4-monitor-acl) for traffic mirroring */
Router(config)# ipv4 access-list v4-monitor-acl
Router(config-ipv4-acl)# 10 permit udp 20.1.1.0 0.0.0.255 eq 10 any
Router(config-ipv4-acl)# 20 permit udp 30.1.1.0 0.0.0.255 eq 20 any
Router(config-ipv4-acl)# exit
Router(config)# commit

/*Create a SPAN IPv6 ACL (v6-monitor-acl) for traffic mirroring */
Router(config)# ipv6 access-list v6-monitor-acl
Router(config-ipv6-acl)# 10 permit ipv6 host 120:1:1::1 host 130:1:1::1
Router(config-ipv6-acl)# exit

/* Apply the traffic monitoring to SPAN source interface */
Router(config)# interface HundredGigE0/0/0/12
Router(config-if)# monitor-session mon1 ethernet direction rx-only

```

```
Router(config-if)# acl ipv4 v4-monitor-acl
Router(config-if)# acl ipv4 v6-monitor-acl!
```



Note The `capture` keyword which is required for Security ACL for SPAN to work, is optional for SPAN ACL.

Use the `show access-lists [ipv4 | ipv6] acl-name hardware ingress span [detail | interface | location | sequence | verify] location x` command to display ACL information:

```
Router# show access-lists ipv4 v4span1 hardware ingress span interface bundle-Ether 100
location 0/3/cpu0
ipv4 access-list v4span1
10 permit ipv4 host 51.0.0.0 host 101.0.0.0
20 permit ipv4 host 51.0.0.1 host 101.0.0.1
30 permit ipv4 host 51.0.0.2 any
40 permit ipv4 any host 101.0.0.3
50 permit ipv4 51.0.1.0 0.0.0.255 101.0.1.0 0.0.0.255
60 permit ipv4 51.0.2.0 0.0.0.255 101.0.2.0 0.0.0.255 precedence critical
```

Troubleshooting ACL-Based Traffic Mirroring

Take note of these configuration issues:

- Even when the system configures the **acl** command on the source mirroring port, if the ACL configuration command does not use the **capture** keyword, the system does not mirror traffic.
- If the ACL configuration uses the **capture** keyword, but you have not configured the **acl** command on the source port, the system mirrors the traffic, but does not apply access list configuration.

This example shows both the **capture** keyword in the ACL definition and the **acl** command that is configured on the interface:

```
/* Create an IPv4 ACL (TM-ACL) for traffic mirroring */
Router(config)# ipv4 access-list TM-ACL
Router(config-ipv4-acl)# 10 permit udp 10.1.1.0 0.0.0.255 eq 10 any capture
Router(config-ipv4-acl)# 20 permit udp 10.1.1.0 0.0.0.255 eq 20 any
```

```
Apply the traffic monitoring to interface
Router(config)#interface HundredGigE0/0/0/12
Router(config-if)# monitor-session mon1 ethernet direction rx-only port-only acl
Router(config-if)# ipv4 access-group TM-ACL ingress
```

Flexible CLI for ERSPAN

Starting with Cisco IOS XR Software Release 7.0.14, ERSPAN can be configured using flexible CLI. This CLI is a single configuration object containing all the properties of an ERSPAN session, tunnel properties, and the list of source interfaces, which can be easily removed and re-added. Flexible CLI minimises risk of user error and promotes operational simplicity.

Configure a flexible CLI group in ERSPAN containing:

- Global ERSPAN session configuration
- Tunnel interface configuration
- ERSPAN source attachment configuration, applied to a regexp of interface names



Note The flexible CLI group contains only the session and interface properties. The session and interface objects themselves must be created in the configuration as usual.

The following example shows a global flexible CLI configuration:

```
group erspan-group-foo
  monitor-session 'foo' ethernet /* Global configuration */
  destination interface tunnel-ip0
  !
  interface 'tunnel-ip0' /* Tunnel interface configuration */
  tunnel tos 10
  tunnel mode gre ipv4
  tunnel source 10.10.10.1
  tunnel destination 20.20.20.2
  !
  interface 'GigabitEthernet0/0/0/[0-3]' /* Interface configuration */
  monitor-session foo ethernet
  !
end-group
```

To enable all ERSPAN configurations, execute `apply-group erspan-group-foo` command. To disable ERSPAN configuration, delete this command.



Note The following three keywords are regular expressions and must be quoted:

- Definition of session name (example: `foo`)
- Definition of tunnel name (example: `tunnel-ip0`)
- Set of source interface names (example: `GigabitEthernet0/0/0/[0-3]`)

Use the `show running-config inheritance` command to view the final configuration after the group is expanded, and the `show monitor-session status` to check the operational state of ERSPAN session.



Note Starting from Release 7.3.3, when a combination of IP-in-IP decap and GRE ERSPAN tunnels are in use, resource utilization of IP-in-IP decap tunnels is accounted. However, resource utilization of ERSPAN GRE tunnels is not accounted in the *Total In Use* counter of `show controllers npu resources sipidxtbl location all` command output, but the *OOB State* would display *RED* if the total number of IP-in-IP decap and ERSPAN GRE tunnels reach 15.

Attaching the Configurable Source Interface

Step 1 `configure`

Example:

```
RP/0/RP0/CPU0# configure
```

Enters global configuration mode.

Step 2 **interface** *type number*

Example:

```
RP/0/RP0/CPU0(config)# interface HundredGigE 0/1/0/10/0/1/0
```

Enters interface configuration mode for the specified source interface. The interface number is entered in *rack/slot/module/port* notation. For more information about the syntax for the router, use the question mark (?) online help function.

Step 3 **ipv4 access-group** *acl-name* {**ingress** | **egress**}

Example:

```
RP/0/RP0/CPU0(config-if)# ipv4 access-group acl1 ingress
```

Controls access to an interface.

Step 4 **monitor-session** *session-name* **ethernet direction rx-only** **port-level**

Example:

```
RP/0/RP0/CPU0(config-if)# monitor-session mon1 ethernet direction rx-only port-level acl
RP/0/RP0/CPU0(config-if-mon)#
```

Attaches a monitor session to the source interface and enters monitor session configuration mode.

Note **rx-only** specifies that only ingress traffic is replicated.

Step 5 **acl**

Example:

```
RP/0/RP0/CPU0(config-if-mon)# acl
```

Specifies that the traffic mirrored is according to the defined ACL.

Note If an ACL is configured by name then this overrides any ACL that may be configured on the interface.

Step 6 **exit**

Example:

```
RP/0/RP0/CPU0(config-if-mon)# exit
RP/0/RP0/CPU0(config-if)#
```

Exits monitor session configuration mode and returns to interface configuration mode.

Step 7 **end** or **commit**

Example:

```
RP/0/RP0/CPU0(config-if)# end
```

or


```
RP/0/RP0/CPU0(config-if)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

```
Uncommitted changes found, commit them before exiting (yes/no/cancel)?  
[cancel]:
```

- Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
 - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
 - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 8 **show monitor-session [session-name] status [detail] [error]**

Example:

```
RP/0/RP0/CPU0# show monitor-session status
```

Displays information about the monitor session.

Introduction to ERSPAN Rate Limit

With ERSPAN rate limit feature, you can monitor traffic flow through any IP network. This includes third-party switches and routers.

ERSPAN operates in the following modes:

- ERSPAN Source Session – box where the traffic originates (is SPANned).
- ERSPAN Termination Session or Destination Session – box where the traffic is analyzed.

This feature provides rate limiting of the mirroring traffic. With rate limiting, you can limit the amount of traffic to a specific rate, which prevents the network and remote ERSPAN destination traffic overloading. Be informed, if the rate-limit exceeds then the system may cap or drop the monitored traffic.

You can configure the QoS parameters on the traffic monitor session.

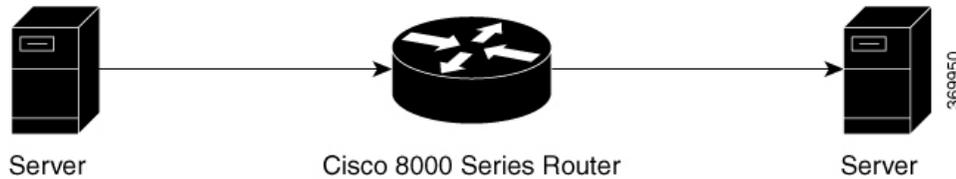
- Traffic Class (0 through 7)
 - Traffic class 0 has the lowest priority and 7 the highest.
 - The default traffic class is the same as that of the original traffic class.

Benefits

With ERSPAN rate limit feature, you can limit the mirrored traffic and use the mirrored traffic for data analysis.

Topology

Figure 4: Topology for ERSPAN Rate Limit



The encapsulated packet for ERSPAN is in ARPA/IP format with GRE encapsulation. The system sends the GRE tunneled packet to the destination box identified by an IP address. At the destination box, SPAN-ASIC decodes this packet and sends out the packets through a port. ERSPAN rate limit feature is applied on the router interface to rate limit the monitored traffic.

The intermediate switches carrying ERSPAN traffic from source session to termination session can belong to any L3 network.

Configure ERSPAN Rate Limit

Use the following steps to configure ERSPAN rate limit:

```

monitor-session ERSPAN ethernet
destination interface tunnel-ip1
!

RP/0/RP0/CPU0:pyke-008#sh run int tunnel-ip 1

interface tunnel-ip1
ipv4 address 4.4.4.1 255.255.255.0
tunnel mode gre ipv4
tunnel source 20.1.1.1
tunnel destination 20.1.1.2
!

RP/0/RP0/CPU0:pyke-008#sh run int hundredGigE 0/0/0/16

interface HundredGigE0/0/0/16
ipv4 address 215.1.1.1 255.255.255.0
ipv6 address 3001::2/64
monitor-session ERSPAN ethernet direction rx-only port-level
  acl
!
ipv4 access-group ACL6 ingress
  
```

Running Configuration

```

!!A traffic class needs to be configured under the monitor session.
monitor-session mon2 ethernet
destination interface tunnel-ip30
traffic class 5
  
```

A shaper needs to be configured for this traffic class:

```

policy-map m8
  
```

```
class TC1
  bandwidth percent 11
!
class TC2
  bandwidth percent 12
!
class TC3
  bandwidth percent 13
!
class TC4
  bandwidth percent 14
!
class TC5
  shape average percent 15
!
class TC6
  bandwidth percent 16
!
class TC7
  bandwidth percent 17
```

This policy-map has to be installed on the interface over which the mirrored traffic is sent in the egress direction:

```
interface TenGigE0/6/0/9/0
service-policy output m8
```

Verification

```
RP/0/RP0/CPU0:ios#show monitor-session FOO status detail
Wed May 2 15:14:05.762 UTC
Monitor-session FOO
Destination interface tunnel-ip100
Source Interfaces
-----
TenGigE0/6/0/4/0
Direction: Both
Port level: True
ACL match: Disabled
```

Introduction to File Mirroring

Prior to Cisco IOS XR Software Release 7.2.1 7.0.14, the router did not support file mirroring from active RP to standby RP. Administrators had to manually perform the task or use EEM scripts to sync files across active RP and standby RP. Starting with Cisco IOS XR Software Release 7.2.1 7.0.14, file mirroring feature enables the router to copy files or directories automatically from `/harddisk:/mirror` location in active RP to `/harddisk:/mirror` location in standby RP or RSP without user intervention or EEM scripts.

Two new CLIs have been introduced for the file mirroring feature:

- **mirror enable**

The `/harddisk:/mirror` directory is created by default, but file mirroring functionality is only enabled by executing the `mirror enable` command from configuration terminal. Status of the mirrored files can be viewed with `show mirror status` command.

- **mirror enable checksum**

The `mirror enable checksum` command enables MD5 checksum across active to standby RP to check integrity of the files. This command is optional.

Limitations

The following limitations apply to file mirroring:

- Supported only on Dual RP systems.
- Supports syncing only from active to standby RP. If files are copied into standby `/harddisk:/mirror` location, it won't be synced to active RP.
- A slight delay is observed in `show mirror` command output when mirror checksum configuration is enabled.
- Not supported on multichassis systems.

Configure File Mirroring

File mirroring has to be enabled explicitly on the router. It is not enabled by default.

```
RP/0/RSP0/CPU0:router#show run mirror
```

```
Thu Jun 25 10:12:17.303 UTC
mirror enable
mirror checksum
```

Following is an example of copying running configuration to `harddisk:/mirror` location:

```
RP/0/RSP0/CPU0:router#copy running-config harddisk:/mirror/run_config
Wed Jul  8 10:25:51.064 PDT
Destination file name (control-c to abort): [/mirror/run_config]?
Building configuration..
32691 lines built in 2 seconds (16345)lines/sec
[OK]
```

Verification

To verify the syncing of file copied to mirror directory, use the `show mirror` command.

```
RP/0/RSP0/CPU0:router#show mirror
Wed Jul  8 10:31:21.644 PDT
% Mirror rsync is using checksum, this show command may take several minutes if you have
many files. Use Ctrl+C to abort
MIRROR DIR: /harddisk:/mirror/
% Last sync of this dir ended at Wed Jul  8 10:31:11 2020
Location      |Mirrored |MD5 Checksum                |Modification Time
-----
run_config |yes      |76fc1b906bec4fe08ecda0c93f6c7815 |Wed Jul  8 10:25:56 2020
```

If checksum is disabled, `show mirror` command displays the following output:

```
RP/0/RSP0/CPU0:router#show mirror
Wed Jul  8 10:39:09.646 PDT
MIRROR DIR: /harddisk:/mirror/
% Last sync of this dir ended at Wed Jul  8 10:31:11 2020
Location      |Mirrored |Modification Time
-----
run_config |yes      |Wed Jul  8 10:25:56 2020
```

If there is a mismatch during the syncing process, use `show mirror mismatch` command to verify.

```
RP/0/RP0/CPU0:router# show mirror mismatch
Wed Jul  8 10:31:21.644 PDT
MIRROR DIR: /harddisk:/mirror/
% Last sync of this dir ended at Wed Jul  8 10:31:11 2020
```

Location	Mismatch Reason	Action Needed
test.txt	newly created item.	send to standby

Introduction to Local SPAN

Local SPAN Overview

Local SPAN is the most basic form of traffic mirroring. In Local SPAN, both mirror source and mirror destination interfaces are present on the same router.

Local SPAN Supported Capabilities

The following capabilities are supported for Local SPAN:

- Only ingress traffic.
- The destination interface can only be an L2 or L3 physical main interface.
- The following interfaces are configured as sources for a Local SPAN session:
 - L3 physical main and sub-interface and bundle main and sub-interface.
 - L2 ethernet interfaces: Ethernet Flow Point (EFP) and trunk
 - BVI interface
- The following types of traffic are mirrored Local SPAN:
 - IPv4, IPv6, and MPLS
 - IP-in-IP
- Extended ACL to reduce mirrored traffic throughput
- Traffic shaping on the destination interface
- Session statistics. There's one counter for all types of traffic, that is, IPv4, IPv6, and MPLS.
- Up to four Local SPAN sessions. This session number is shared between ERSPAN, Local SPAN, and SPAN to File features.
- Up to 1000 source interfaces

Local SPAN Restrictions

The following are the restrictions for Local SPAN:

- Egress mirroring isn't supported
- The physical interface used as destination can't be a bundle member link
- GRE tunnel isn't supported as source interface and destination interface

- Port-level monitoring isn't supported
- Per-source interface mirroring statistics isn't supported. However, SPAN session statistics are supported. The session statistics would contain total number of packets mirrored by the session.
- A destination interface can't be a mirrored source interface and vice versa.
- ACL for Local SPAN is only applied in ingress direction.
- If ACL keyword is present in monitor-session configuration for an interface but no ACL is applied to that interface, traffic packets won't be mirrored
- No ACL support for MPLS traffic
- No support for NetFlow or sFlow configuration on the same interface which has Local SPAN session already configured.

Configuring Local SPAN

Configuring Local SPAN consists of 2 parts:

1. Creating a local SPAN session

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config)#monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#destination interface HundredGigE0/1/0/0
RP/0/RP0/CPU0:router(config-mon)#commit
RP/0/RP0/CPU0:router(config-mon)#end
RP/0/RP0/CPU0:router#
```

2. Attaching the SPAN session to an interface

```
RP/0/RP0/CPU0:router(config-mon)#interface HundredGigE0/1/0/2
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)# no shut
RP/0/RP0/CPU0:router(config-if)#!
RP/0/RP0/CPU0:router(config-if)#
RP/0/RP0/CPU0:router(config-if)#interface Bundle-Ether1
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:router(config-if-mon)# no shutdown
RP/0/RP0/CPU0:router(config-if)#!
RP/0/RP0/CPU0:router(config-if)#
RP/0/RP0/CPU0:monitor(config-if)#
RP/0/RP0/CPU0:monitor(config-if)#interface HundredGigE0/1/0/14.100
RP/0/RP0/CPU0:monitor(config-subif)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:monitor(config-if-mon)# no shut
RP/0/RP0/CPU0:monitor(config-subif)#!
RP/0/RP0/CPU0:monitor(config-subif)#
RP/0/RP0/CPU0:monitor(config-subif)#interface Bundle-Ether1.1
RP/0/RP0/CPU0:monitor(config-subif)# monitor-session mon1 ethernet direction rx-only
RP/0/RP0/CPU0:monitor(config-if-mon)# no shut
RP/0/RP0/CPU0:monitor(config-subif)#!
RP/0/RP0/CPU0:monitor(config-subif)#commit
```

Verification

```
RP/0/RP0/CPU0:router#show monitor-session status
Monitor-session mon1
Destination interface HundredGigE0/1/0/0
=====
Source Interface      Dir      Status
```

```

-----
Hu0/1/0/2           Rx    Operational
Hu0/1/0/14.100     Rx    Operational
BE1                 Rx    Operational
BE1.1               Rx    Operational

```

Execute the `show monitor-session status internal` command for session statistics:

```

RP/0/RP0/CPU0:router#show monitor-session status internal
Thu Aug 13 20:05:23.478 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon1 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface HundredGigE0/1/0/0 (0x00800190)
          Last error: Success
0/1/CPU0: Destination interface HundredGigE0/1/0/0 (0x00800190)
0/RP0/CPU0: Destination interface HundredGigE0/1/0/0 (0x00800190)
Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/1/CPU0: Name 'mon1', destination interface HundredGigE0/1/0/0 (0x00800190)
Platform, 0/1/CPU0:

    Monitor Session ID: 1
    Monitor Session Packets: 32
    Monitor Session Bytes: 4024

0/2/CPU0: Name 'mon1', destination interface HundredGigE0/1/0/0 (0x00800190)
Platform, 0/2/CPU0:

    Monitor Session ID: 1
    Monitor Session Packets: 0
    Monitor Session Bytes: 0

```

Local SPAN with ACL

Local SPAN with ACL is used to filter and mirror ingress traffic. Only Access Control Entries (ACEs) with `capture` keyword are considered for mirroring. Both permit and deny packets are captured if the ACE contains `capture` keyword. Per interface, only one IPv4 ingress ACL and one IPv6 ingress ACL is allowed.

Configuring Local SPAN with ACL

Use the following configuration to enable local SPAN with IPv4 ACLs:

1. Configure ACLs for traffic mirroring.

```

Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 25.0.0.0 0.0.0.255 any capture
Router(config-ipv4-acl)# 20 permit ipv4 20.0.0.0 0.0.0.255 any
Router(config-ipv4-acl)# 30 permit ipv4 131.1.1.0 0.0.0.255 any capture
Router(config-ipv4-acl)# 40 permit ipv4 191.1.1.0 0.0.0.255 any capture

```

2. Apply the traffic monitoring to an interface.

```

Router(config)# interface HundredGigE0/1/0/2
Router(config-if)# ipv4 address 131.1.1.2 255.255.255.0
Router(config-if)# monitor-session mon1 ethernet direction rx-only port-level
Router(config-if-mon)# acl
Router(config-if-mon)# ipv4 access-group acl1 ingress

```

Verification

```

RP/0/RP0/CPU0:ios#show running-config ipv4 access-list acl1
Thu Aug 13 20:22:54.388 UTC

```

```

ipv4 access-list acl1
10 permit ipv4 22.0.0.0 0.0.0.255 any capture
20 permit ipv4 20.0.0.0 0.0.0.255 any
30 permit ipv4 131.1.1.0 0.0.0.255 any capture
40 deny ipv4 181.1.1.0 0.0.0.255 any capture
!

```

Use the following configuration to enable local SPAN with IPv6 ACLs:

1. Configure ACLs for traffic mirroring.

```

Router(config)# ipv6 access-list acl2
Router(config-ipv6-acl)# 10 permit ipv6 10:1:1::2/64 any capture
Router(config-ipv6-acl)# 20 permit ipv6 10:1:1::3/64 any
Router(config-ipv6-acl)# 30 permit ipv6 10:1:1::4/64 any capture

```

2. Apply the traffic monitoring to an interface.

```

Router(config)# interface HundredGigE0/1/0/3
Router(config-if)# ipv6 address 10:1:1::5/64
Router(config-if)# monitor-session mon2 ethernet direction rx-only port-level
Router(config-if-mon)# acl
Router(config-if-mon)# ipv6 access-group acl2 ingress

```

Verification

```

RP/0/RP0/CPU0:ios#show running-config ipv6 access-list acl2
Thu Aug 14 20:22:54.388 UTC
ipv6 access-list acl2
10 permit ipv6 10:1:1::2/64 any capture
20 permit ipv6 10:1:1::3/64 any
30 permit ipv6 10:1:1::4/64 any capture
!

```

Local SPAN Rate Limit

Local SPAN rate limiting takes place at the session level and not at source interface level. For rate limiting, local SPAN session should configure a traffic class. This traffic class is used to shape traffic on an egress interface. A QoS policy is applied to the egress interface over which mirrored traffic is sent.

Example for Local SPAN Rate Limit Configuration

```

Router# monitor-session mon2 ethernet
destination interface HundredGigE0/1/0/19
traffic-class 5

class-map match-any TC5
match traffic-class 5
end-class-map

policy-map shape-foo
class TC5 /* This has to match the class that was configured on monitor session */
shape average percent 15
class class-default

interface HundredGigE0/1/0/19 /* This is the egress interface over which mirrored packets
are sent */
service-policy output shape-foo

```


Traffic Mirroring with DSCP

Differentiated Service Code Point (DSCP) value of Differentiated Services (DS) field in IP packet is used to classify the traffic in the network. DS field formerly known as Type of Service (ToS). You can set the DSCP value in the six most significant bits of the differentiated services (DS) field of the IP header, thereby giving $2^6 = 64$ different values (0 to 63). These six bits affect the Per Hop Behavior (PHB) and hence affects how a packet is moved forward. The default value of DSCP is zero (0). DSCP was defined under RFC 2474.

Following the principle of traffic classification, DSCP places a particular packet into a limited number of traffic classes. Similarly, the router is also informed about the DSCP values and the router can prioritize the packet in traffic flow.

Refer the table to know more about the service class names defined in [RFC 2474](#).

Table 2: DSCP, DS, and ToS values

DSCP Value in Decimal	DS Binary	DS Hex	DSCP Name	DS/ToS Value	Service Class
0	000000	0x00	DF/CS0	0	Standard
-	-	-	none	2	
1	000001	0x01	None	4	
1	000001	0x01	LE	4	Lower-effort
2	000010	0x02	None	8	
4	000100	0x04	None	16	
8	001 000	0x08	CS1	32	Low-priority data
10	001 010	0x0a	AF11	40	High-throughput data
12	001 100	0x0c	AF12	48	High-throughput data
14	001 110	0x0e	AF13	56	High-throughput data
16	010 000	0x10	CS2	64	OAM
18	010 010	0x12	AF21	72	Low-latency data
20	010 100	0x14	AF22	80	Low-latency data
22	010 010	0x16	AF23	88	Low-latency data
24	011 000	0x18	CS3	96	Broadcast video
26	011 000	0x1a	AF31	104	Multimedia streaming
28	011 100	0x1c	AF32	112	Multimedia streaming
30	011 110	0x1e	AF33	120	Multimedia streaming
32	100 000	0x20	CS4	128	Real-time interactive
34	100 010	0x22	AF41	136	Multimedia conferencing
36	100 100	0x24	AF42	144	Multimedia conferencing

38	100 110	0x26	AF43	152	Multimedia conferencing
40	101 000	0x28	CS5	160	Signaling(IP telephony, etc)
44	101 100	0x2c	Voice-admit	176	
46	101 110	0x2e	EF	184	Telephony
48	110 000	0x30	CS6	192	Networkrouting control
56	111 000	0x38	CS7	224	“reserved”

DSCP Bitmask to Filter Ingress ERSPAN Traffic

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
DSCP Bitmask to Filter Ingress ERSPAN Traffic	Release 7.5.4	<p>You can now mirror multiple traffic flows for matched Differentiated Service Code Point (DSCP) value of IP header on the Encapsulated remote SPAN (ERSPAN). The matched DSCP value is based on the DSCP value and the bitmask configured in Access Control List (ACL) rule.</p> <p>Earlier, you could monitor single traffic flow by setting the RFC 4594 defined DSCP values in the GRE tunnel header.</p> <p>This feature introduces the following changes:</p> <ul style="list-style-type: none"> • CLI: deny (IPv4), deny (IPv6), permit (IPv4), and permit (IPv6) are modified to include new keyword bitmask. • YANG DATA Model: New XPaths for Cisco-IOS-XR-um-ipv4-access-list-cfg and Cisco-IOS-XR-um-ipv6-access-list-cfg (see Github, YANG Data Models Navigator).

Starting Release 7.5.4, You can configure an ACL rule with DSCP bitmask on the ERSPAN GRE tunnels to mirror specific traffic flows.

Without ACL rule, ERSPAN mirrors all the traffic on the incoming port. When ACL is configured with DSCP and DSCP mask on the ERSPAN, ERSPAN mirrors the traffic whose DSCP value lies within the combination of DSCP value and the specified mask.

A DSCP value is mapped to a single traffic class as per the defined value in [RFC2474](#). Masking the DSCP value in ACL rule allows to mirror multiple traffic flows. DSCP value and mask operate similar to IPv4 address and mask.

Configure DSCP Bitmask to Filter Ingress ERSPAN Traffic

To configure DSCP bitmask, use the `bitmask` option along with the `dscp` option while configuring the ACL.

Configuration Example for IPv4

This example shows how you can configure DSCP bitmask on ingress ERSPAN for IPv4 traffic.

```
/*configure the ACL*/
Router# config
Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
Router(config-ipv4-acl)# commit
Router(config-ipv4-acl)# exit

/* Perform the following configurations to attach the created ACL to an interface*/
Router(config)# interface HundredGigE0/0/0/6
Router(config-if)# ipv4 address 192.0.2.51 255.255.255.0

/* Monitor the ingress ACL applied and DSCP masked IPv4 traffic on ERSPAN*/
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv4 acl1
Router(config-if)# commit
```

Running Configuration

```
Router(config)# show running-config ipv4 access-list
ipv4 access-list acl1
 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
!

interface HundredGigE0/0/0/6
  ipv4 address 192.0.2.51 255.255.255.0
  monitor-session TEST ethernet direction rx-only port-level acl ipv4 acl1
!
!
```

Configuration Example for IPv6

This example shows how you can configure DSCP bitmask on ingress ERSPAN for IPv6 traffic.

```
/*configure the ACL*/
Router# config
Router(config)# ipv6 access-list acl1
Router(config-ipv6-acl)# 10 permit ipv6 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit

/* Perform the following configurations to attach the created ACL to an interface*/
Router(config)# interface HundredGigE 0/0/10/3
Router(config-if)# ipv6 address 2001:DB8::1/32

/* Monitor the ingress ACL applied and DSCP masked IPv4 traffic on ERSPAN*/
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
Router(config-if)# commit
```

Running Configuration

```
Router(config)# show running-config ipv6 access-list
ipv6 access-list acl1
 10 permit ipv6 acl1 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
!

interface HundredGigE0/0/10/3
  ipv6 address 2001:db8::1/32
  monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
!
!
```

SPAN to File

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
SPAN to File Support for TX and RX	Release 7.5.3	<p>With this feature, the ability to capture the packet in TX direction along with the ability to store the capture on the file is supported.</p> <p>You can now capture the packet in the TX direction and store the capture on the file. Earlier, you could only capture or mirror the traffic in the RX direction. You now have the flexibility to choose TX, RX, or both directions.</p> <p>You can now capture and analyze the outgoing (TX) packets.</p>
Partial Packet Capture Ability for SPAN-to-File (RX)	Release 7.5.3	<p>With this feature, you can perform partial packet capture in the RX direction.</p> <p>Earlier, the ability for entire packet capture was available in the TX direction only, now you can choose entire or partial packet capture in the Rx direction also.</p> <p>Here, partial packet capture is also known as truncation.</p>
SPAN to File - PCAPng File Format	Release 7.3.1	<p>PCAPng is the next generation of packet capture format that contains a dump of data packets captured over a network and stored in a standard format.</p> <p>The PCAPng file contains different types of information blocks, such as the section header, interface description, enhanced packet, simple packet, name resolution, and interface statistics. These blocks can be used to rebuild the captured packets into recognizable data.</p> <p>The PCAPng file format:</p> <ul style="list-style-type: none"> • Provides the capability to enhance and extend the existing capabilities of data storage over time • Allows you to merge or append data to an existing file. • Enables to read data independently from network, hardware, and operating system of the machine that made the capture.

SPAN to File is an extension of the pre-existing SPAN feature that allows network packets to be mirrored to a file instead of an interface. This helps in the analysis of the packets at a later stage. The file format is PCAP, which helps that data to be used by tools, such as tcpdump or Wireshark.



Note A maximum of 100 source ports are supported across the system. Individual platforms may support lower numbers. All the SPAN sessions are configured under the Ethernet class. At any given time, the system supports four SPAN to File sessions.

When you configure a file as a destination for a SPAN session, the system creates buffer on each node to which the network packets are logged. The buffer is for all packets on the node regardless of which interface they are from. That is, multiple interfaces can provide packets to the same buffer. The system deletes the buffer when the session configuration is removed. Each node writes a file on the active RP, which contains the node ID of the node on which the buffer was located.

The minimum buffer size is 1KB. The maximum buffer size is 1000KB and default buffer size is 2KB.

If multiple interfaces are attached to a session, then interfaces on the same node are expected to have their packets sent to the same file. Bundle interfaces can be attached to a session with a file destination, which is similar to attaching individual interfaces.

From Cisco IOS XR Software Release 7.5.3 onwards, the capture of all the outgoing packets from the router is supported.

Earlier to Cisco IOS XR Software Release 7.5.3, there was no functionality which enables to capture the payload of packets coming from your customers for security reasons.

Limitations and Restrictions for SPAN to File

- Only incoming packet mirroring on the source interface is supported. Outgoing mirrored packets cannot be dumped to the file.

However, from Cisco IOS XR Software Release 7.5.3 onwards, there are no restrictions.

- SPAN ACLs can only be applied in ingress direction only. Hence, ACLs for SPAN to File can only be applied in ingress direction only.
- ACL on MPLS traffic is not supported.
- MPLS over GRE traffic is supported, however, GRE interfaces cannot be configured as source interfaces.
- Packet truncation applies for SPAN to File and ERSPAN interfaces only. If you change the destination to Local SPAN, then an **ios_msg** is displayed as a warning. The entire packet is mirrored after this message is displayed.

Example: The Partial Packet Capture feature is not supported by Local SPAN. The entire Packet will be mirrored.

- Packet truncation is per monitor session.
- Currently, truncation per interface is not supported.
- For outgoing (TX) SPAN to File, Security ACL is not supported.
- For outgoing (TX) SPAN to File, only transit traffic is mirrored.
Self-originating traffic cannot be mirrored.

Supported capabilities for SPAN to File

- The ability to mirror outgoing traffic and punt it to the CPU across all NPU versions.
- Ability to mirror outgoing IPv4, IPv6, and MPLS traffic to file.
- Ability to mirror outgoing traffic across all types of L3 interfaces, including physical, sub, bundle, and bundle sub interfaces
- Ability to mirror outgoing traffic across L2 or BVI interfaces.

- Ability to enable the new SPAN to File truncation configuration for both RX and TX direction. You can specify the `both` keyword to enable RX and TX mirroring on a single source interface.

See [Configuring SPAN to File for Truncation and Direction, on page 32](#)

- Ability to configure a different truncation size on each monitor session.
- Ability to configure SPAN to File mirroring packet truncation size from 1 to 10000. If you try to configure a value out of the range, the configuration will not accept it and displays an error message.
- Ability to change the truncation size, when packet collecting has stopped. Removing or re-adding the monitor session is not required.
- Ability to change the truncation size during packet collecting ON. Not required to stop the monitor session.
- The entire packet is mirrored by default, without the mirror first (truncation size) configuration.

Also, if the packet size is less than the configured truncation size, the entire packet is mirrored.

Action Commands for SPAN to File

Action commands allows you to start and stop network packet collection. You can run the action commands on sessions where the destination is a file. The action command auto completes names of the globally configured SPAN to File sessions. The following table provides more information on action commands.

Table 5: Action Commands for SPAN to File

Action	Command	Description
Start	<code>monitor-session <name> packet-collection start</code>	Use this command to start writing packets for the specified session to the configured buffer.
Stop	<code>monitor-session <name> packet-collection stop [discard-data write directory <dir> filename <filename>]</code>	Use this command to stop writing packets to the configured buffer. If you specify the <code>discard-data</code> option, the system clears the buffer. Whereas if you specify the <code>write</code> option, the system writes the buffer to disk before clearing. When you must write buffer to disk, you must save the file in a <code>.pap</code> format at this location, <code><directory>/<node_id>/<filename>.pcap</code> . If you add a <code>.pcap</code> extension while specifying the filename, the system removes <code>.pcap</code> so that the extension is not added twice.

Configuring SPAN to File

Use the following command to configure SPAN to File:

```
monitor-session <name> [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
  destination file [size <kbytes>] [buffer-type linear]
```

The `monitor-session <name> [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]` part of the command creates a monitor-session with the specified name and class and is a pre-existing chain point from the current SPAN feature. The `destination file [size <kbytes>] [buffer-type linear]` part of the command adds a new “file” option to the existing “destination”.

`destination file` has the following configuration options:

- Buffer size.
- Two types of buffer:
 - Circular: Once the buffer is full, the start is overwritten.
 - Linear: Once the buffer is full, no further packets are logged.



Note The default buffer-type is circular. Only linear buffer is explicitly configurable. Changing any of the parameters (buffer size or type) recreates the session, and clears any buffers of packets.

All configuration options which are applied to an attachment currently supported for other SPAN types should also be supported by SPAN to file. This may include:

- ACLs
- Write only first X bytes of packet.
- In Cisco IOS XR Release 7.5.3, truncation per global session is supported and not per interface.



Note These options are implemented by the platform when punting the packet.

Once a session has been created, then interfaces may be attached to it using the following configuration:

```
interface GigabitEthernet 0/0/0/0
  monitor-session <name> [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
```

The attachment configuration is unchanged by SPAN to File feature.

Configuration Examples

To configure a `mon1` monitor session, use the following commands:

```
monitor-session mon1 ethernet
  destination file size 230000
  !
```

In the above example, omitting the `buffer-type` option results in default circular buffer.

To configure a `mon2` monitor session, use the following commands:

```
monitor-session mon2 ethernet
  destination file size 1000 buffer-type linear
  !
```

To attach monitor session to a physical or bundle interface, use the following commands:

```
RP/0/RSP0/CPU0:router#show run interface Bundle-Ether 1
Fri Apr 24 12:12:59.348 EDT
interface Bundle-Ether1
monitor-session ms7 ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
[direction {rx-only|tx-only|both[SW(1) ]} [port-level]
acl [<acl_name>]!
```

Running Configuration

```
!! IOS XR Configuration 7.1.1.124I
!! Last configuration change at Tue Nov 26 19:29:05 2019 by root
!
hostname OC
logging console informational
!
monitor-session mon1 ethernet
  destination file size 230000 buffer-type circular
!
monitor-session mon2 ethernet
  destination file size 1000 buffer-type linear

!
interface Bundle-Ether1
monitor-session ms7 ethernet
  direction rx-only
end
```

Verification

To verify packet collection status:

```
RP/0/RP0/CPU0:router#show monitor-session status
Monitor-session mon1
Destination File - Packet collecting
=====
Source Interface      Dir.      Status
-----
Hu0/9/0/2              Rx        Operational

Monitor-session mon2
Destination File - Packet collecting
=====
Source Interface      Dir      Status
-----
BE2.1.                Rx        Operational
```

If packet collection is not active, the following line is displayed:

```
Monitor-session mon2
Destination File - Not collecting
```

Configuring SPAN to File for Truncation and Direction

Configuring SPAN to File for Truncation

Use the **mirror first** command in monitor session configuration mode to create a SPAN to File monitor session for mirroring the packets with truncation enabled:

```
monitor-session <name> [ethernet]
destination file [size <kbytes>] [buffer-type linear|circular]
mirror first <number>
```

Once a session has been created, then interfaces may be attached to it using the following configuration:


```
interface <>
  monitor-session session-name ethernet direction rx-only|tx-only|both | acl [acl_name]
```

Configuration Examples

To configure a mon1 monitor session, use the following commands:

```
monitor-session mon1 ethernet
  destination file
  mirror first 128
!
```

Configuring SPAN to File for Direction

Use the following command to create a SPAN to File monitor session for mirroring the packets:

```
monitor-session mon2 ethernet
  destination file
!
```

Attach the session which has been created to the interfaces using the following configuration:

```
interface <>
  monitor-session session-name ethernet direction rx-only|tx-only|
acl [acl_name]
```

Running Configuration for all

```
monitor-session mon3 ethernet
destination file
!

interface Hu0/9/0/2
monitor-session mon1 ethernet
direction rx-only
!

interface bundle-ether1
monitor-session mon2 ethernet
direction tx-only
!

interface bundle-ether2.1
monitor-session mon3 ethernet
direction both
end
```

Verification

The **show monitor-session status** displays the direction.

```
Router#show monitor-session status
Monitor-session mon1
Destination File - Packet collecting
=====
Source Interface      Dir      Status
-----
Hu0/9/0/2             Rx       Operational
Monitor-session mon2
Destination File - Packet collecting
=====
Source Interface      Dir      Status
-----
BE1                   Tx       Operational
Monitor-session mon3
Destination File - Packet collecting
```

```

=====
Source Interface      Dir   Status
-----
BE2.1                 Both  Operational

```

Traffic Mirroring Configuration Examples

This section contains examples of how to configure traffic mirroring:

Viewing Monitor Session Status: Example

This example shows sample output of the `show monitor-session` command with the `status` keyword:

```

RP/0/RP0/CPU0:router# show monitor-session status

Monitor-session cisco-rtpl
Destination interface HundredGigE0/5/0/38
=====
Source Interface  Dir   Status
-----
Gi0/5/0/4        Rx   Operational
Gi0/5/0/17       Rx   Operational

RP/0/RP0/CPU0:router# show monitor-session status detail

Monitor-session sess1
Destination interface is not configured
Source Interfaces
-----
HundredGigE0/0/0/0
Direction: Rx
ACL match: Enabled
Portion: Full packet
Status: Not operational (destination interface not known).
HundredGigE0/0/0/2
Direction: Rx
ACL match: Disabled
Portion: First 100 bytes

RP/0/RP0/CPU0:router# show monitor-session status error

Monitor-session ms1
Destination interface HundredGigE0/2/0/15 is not configured
=====
Source Interface  Dir   Status
-----

Monitor-session ms2
Destination interface is not configured
=====
Source Interface  Dir   Status
-----

```

Monitor Session Statistics: Example

The monitor session statistics is provided in the form of packets and bytes. Use the following command to get the status:

```
RP/0/RP0/CPU0:router# show monitor-session <session_name> status internal
RP/0/RP0/CPU0:Router1#show monitor-session mon2 status internal
Wed Oct 9 19:39:30.402 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.1.1.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set
0/1/CPU0: Destination interface tunnel-ip2 (0x0f000034)
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 2.2.2.2
            Dest IP: 130.1.1.2
            VRF:
            ToS: 0 (copied)
            TTL: 255
            DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/1/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/1/CPU0:

          Monitor Session ID: 1

          Monitor Session Packets: 11
          Monitor Session Bytes: 1764

0/2/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/2/CPU0:

          Monitor Session ID: 1

          Monitor Session Packets: 0
          Monitor Session Bytes: 0
```



Note

- Currently, the system does not allow you to clear these counters.
- The counters are present on the line-card that contains the interface over which the mirrored packets are sent to the ERSPAN session destination.

If required, to clear the counters, delete and recreate the monitor session. Also, clear the counters by performing a Shut/No Shut of the tunnel interface, which triggers a Delete+Create action.

Layer 3 ACL-Based Traffic Mirroring: Example

This example shows how to configure Layer 3 ACL-based traffic mirroring:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session msl
RP/0/RP0/CPU0:router(config-mon)# destination tunnel-ip 1

RP/0/RP0/CPU0:router(config-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE/2/0/11
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group span ingress
RP/0/RP0/CPU0:router(config-if)# monitor-session msl ethernet direction rx-only acl
RP/0/RP0/CPU0:router(config-if-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 access-list span
RP/0/RP0/CPU0:router(config-ipv4-acl)# 5 permit ipv4 any any dscp 5 capture
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
```

Troubleshooting Traffic Mirroring

When you encounter any issue with traffic mirroring, begin troubleshooting by checking the output of the **show monitor-session status** command. This command displays the recorded state of all sessions and source interfaces:

```
Monitor-session sess1
<Session status>
=====
Source Interface  Dir  Status
-----
Gi0/0/0/0        Both <Source interface status>
Gi0/0/0/2        Both <Source interface status>
```

In the preceding example, the line marked as `<Session status>` can indicate one of these configuration errors:

Session Status	Explanation
Session is not configured globally	The session does not exist in global configuration. Check show run command output to ensure that a session with a correct name has been configured.
Destination interface <intf> is not configured	The interface that has been configured as the destination does not exist. For example, the destination interface may be configured to be a VLAN subinterface, but the VLAN subinterface may not have been yet created.

Session Status	Explanation
Destination interface <intf> (<down-state>)	The destination interface is not in Up state in the Interface Manager. You can verify the state using the show interfaces command. Check the configuration to see what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).

The <Source interface status> can report these messages:

Source Interface Status	Explanation
Operational	Everything appears to be working correctly in traffic mirroring PI. Please follow up with the platform teams in the first instance, if mirroring is not operating as expected.
Not operational (Session is not configured globally)	The session does not exist in global configuration. Check the show run command output to ensure that a session with the right name has been configured.
Not operational (destination interface not known)	The session exists, but it either does not have a destination interface specified, or the destination interface named for the session does not exist (for example, if the destination is a sub-interface that has not been created).
Not operational (source same as destination)	The session exists, but the destination and source are the same interface, so traffic mirroring does not work.
Not operational (destination not active)	The destination interface or pseudowire is not in the Up state. See the corresponding <i>Session status</i> error messages for suggested resolution.
Not operational (source state <down-state>)	The source interface is not in the Up state. You can verify the state using the show interfaces command. Check the configuration to see what might be keeping the interface from coming up (for example, a sub-interface needs to have an appropriate encapsulation configured).
Error: see detailed output for explanation	Traffic mirroring has encountered an error. Run the show monitor-session status detail command to display more information.

The **show monitor-session status detail** command displays full details of the configuration parameters, and of any errors encountered. For example:

```
RP/0/RP0/CPU: router#show monitor-session status detail

Monitor-session sess1
  Destination interface is not configured
  Source Interfaces
```

```

-----
HundredGigE0/0/0/0
  Direction: Both
  ACL match: Enabled
  Portion: Full packet
  Status: Not operational (destination interface not known)
HundredGigE0/0/0/2
  Direction: Both
  ACL match: Disabled
  Portion: First 100 bytes
  Status: Not operational (destination interface not known). Error: 'Viking SPAN PD' detected
the 'warning' condition 'PRM connection creation failure'.
Monitor-session foo
Destination next-hop HundredGigE 0/0/0/0
Source Interfaces
-----
HundredGigE 0/1/0/0.100:
  Direction: Both
  Status: Operating
HundredGigE 0/2/0/0.200:
  Direction: Tx
  Status: Error: <blah>

Monitor session bar
No destination configured
Source Interfaces
-----
HundredGigE 0/3/0/0.100:
  Direction: Rx
  Status: Not operational(no destination)

```

Additional Debugging Commands

Here are additional trace and debug commands:

```

RP/0/RP0/CPU0:router# show monitor-session platform trace ?

all    Turn on all the trace
errors Display errors
events Display interesting events

RP/0/RP0/CPU0:router# show monitor-session trace ?

process Filter debug by process

RP/0/RP0/CPU0:router# debug monitor-session platform ?

all    Turn on all the debugs
errors VKG SPAN EA errors
event  VKG SPAN EA event
info   VKG SPAN EA info

RP/0/RP0/CPU0:router# debug monitor-session platform all

RP/0/RP0/CPU0:router# debug monitor-session platform event

RP/0/RP0/CPU0:router# debug monitor-session platform info

RP/0/RP0/CPU0:router# show monitor-session status ?

detail Display detailed output
errors  Display only attachments which have errors

```

```

internal Display internal monitor-session information
|      Output Modifiers

RP/0/RP0/CPU0:router# show monitor-session status

RP/0/RP0/CPU0:router# show monitor-session status errors

RP/0/RP0/CPU0:router# show monitor-session status internal

```

If there is no route to the destination IPv4 address, the status displayed for the monitor session looks like this:

```

RP/0/RP0/CPU0:Router1#show monitor-session mon2 status internal
Wed Oct  9 19:24:06.084 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034) (down)
Last error: Success
Tunnel data:
  Mode: GREoIPv4
  Source IP: 2.2.2.2
  Dest IP: 130.10.10.2
  VRF:
  ToS: 0 (copied)
  TTL: 255
  DFbit: Not set
0/1/CPU0: Destination interface is not configured
Tunnel data:
  Mode: GREoIPv4
  Source IP: 2.2.2.2
  Dest IP: 130.10.10.2
  VRF:
  ToS: 0 (copied)
  TTL: 255
  DFbit: Not set

```

To verify if there is a route to the destination IPv4 address, use the following command:

```

RP/0/RP0/CPU0:Router1#show cef ipv4 130.10.10.2
Wed Oct  9 19:25:12.282 UTC
0.0.0.0/0, version 0, proxy default, default route handler, drop adjacency, internal 0x1001011
 0x0 (ptr 0x8e88d2b8) [1], 0x0 (0x8ea4d0a8), 0x0 (0x0)
Updated Oct  9 19:03:36.068
Prefix Len 0, traffic index 0, precedence n/a, priority 15
  via 0.0.0.0/32, 3 dependencies, weight 0, class 0 [flags 0x0]
  path-idx 0 NHID 0x0 [0x8e2db240 0x0]
  next hop 0.0.0.0/32
  drop adjacency

```

When a route is present, the command used in the previous example displays the following:

```

RP/0/RP0/CPU0:Router1#show cef ipv4 130.10.10.2
Wed Oct  9 19:26:06.141 UTC
130.1.1.0/24, version 20, internal 0x1000001 0x0 (ptr 0x8e88aa18) [1], 0x0 (0x8ea4dc68),
0x0 (0x0)
Updated Oct  9 19:26:02.139
Prefix Len 24, traffic index 0, precedence n/a, priority 3
  via 131.1.1.1/32, HundredGigE0/1/0/2, 2 dependencies, weight 0, class 0 [flags 0x0]
  path-idx 0 NHID 0x0 [0x8f8e2260 0x0]
  next hop 131.10.10.1/32
  local adjacency

```

The show monitor command displays the following:

```

show monitor-session mon2 status internal
Wed Oct  9 19:26:12.405 UTC
Information from SPAN Manager and MA on all nodes:

```

```

Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034)
Last error: Success
Tunnel data:
  Mode: GREoIPv4
  Source IP: 2.2.2.2
  Dest IP: 130.10.10.2
  VRF:
  ToS: 0 (copied)
  TTL: 255
  DFbit: Not set
0/1/CPU0: Destination interface tunnel-ip2 (0x0f000034)
Tunnel data:
  Mode: GREoIPv4
  Source IP: 2.2.2.2
  Dest IP: 130.10.10.2
  VRF:
  ToS: 0 (copied)
  TTL: 255
  DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/1/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/1/CPU0:

  Monitor Session ID: 1

  Monitor Session Packets: 0
  Monitor Session Bytes: 0

0/2/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)
Platform, 0/2/CPU0:

  Monitor Session ID: 1

  Monitor Session Packets: 0
  Monitor Session Bytes: 0

Missing ARP to the next hop to the destination
This condition is detected via this show command:
show monitor-session mon2 status internal

```

After resolving ARP for the next hop, which is done by invoking a ping command to the destination, the show command output displays the following:

```

RP/0/RP0/CPU0:Router1#show monitor-session mon2 status internal
Wed Oct  9 19:32:24.856 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session mon2 (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x0f000034)
Last error: Success
Tunnel data:
  Mode: GREoIPv4
  Source IP: 2.2.2.2
  Dest IP: 130.10.10.2
  VRF:
  ToS: 0 (copied)
  TTL: 255
  DFbit: Not set
0/1/CPU0: Destination interface tunnel-ip2 (0x0f000034)
Tunnel data:
  Mode: GREoIPv4
  Source IP: 2.2.2.2
  Dest IP: 130.10.10.2

```



```
VRF:  
ToS: 0 (copied)  
TTL: 255  
DFbit: Not set
```

```
Information from SPAN EA on all nodes:  
Monitor-session 0x00000001 (Ethernet)  
0/1/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)  
Platform, 0/1/CPU0:
```

```
Monitor Session ID: 1  
Monitor Session Packets: 0  
Monitor Session Bytes: 0
```

```
0/2/CPU0: Name 'mon2', destination interface tunnel-ip2 (0x0f000034)  
Platform, 0/2/CPU0:
```

```
Monitor Session ID: 1  
Monitor Session Packets: 0  
Monitor Session Bytes: 0
```

Where to Go Next

When you have configured an Ethernet interface, you can configure individual VLAN subinterfaces on that Ethernet interface.

For information about modifying Ethernet management interfaces for the shelf controller (SC), route processor (RP), and distributed RP, see the *Advanced Configuration and Modification of the Management Ethernet Interface* later in this document.

For information about IPv6 see the *Implementing Access Lists and Prefix Lists* on

Cisco IOS XR Software module in the *Cisco IOS XR IP Addresses and Services Configuration Guide*.

