



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).

- [Introduction to Traffic Mirroring, on page 1](#)
- [Traffic Mirroring Types, on page 2](#)
- [ERSPAN, on page 3](#)
- [Configure Traffic Mirroring, on page 4](#)
- [Configure Remote Traffic Mirroring, on page 4](#)
- [Attaching the Configurable Source Interface, on page 7](#)
- [Configuring UDF-Based ACL for Traffic Mirroring, on page 9](#)
- [Additional Information on Traffic Mirroring, on page 10](#)
- [Traffic Mirroring Configuration Examples, on page 12](#)
- [Troubleshooting Traffic Mirroring, on page 13](#)
- [Verifying UDF-based ACL, on page 16](#)
- [Introduction to ERSPAN Egress Rate Limit, on page 16](#)

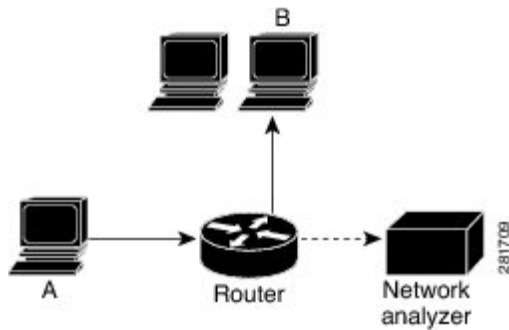
Introduction to Traffic Mirroring

Traffic mirroring, sometimes called port mirroring or Switched Port Analyzer (SPAN), is a Cisco proprietary feature that enables you to monitor network traffic passing in or out of a set of ports. You can then pass this traffic to a destination port on the same router.

Traffic mirroring copies traffic from one or more source ports and 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 flow of traffic on the source interfaces or sub-interfaces. It allows the mirrored traffic to be sent to a destination interface or sub-interface.

For example, you can attach a traffic analyzer to the router and capture Ethernet traffic that is sent by host A to host B.

Figure 1: Traffic Mirroring Operation



When local traffic mirroring is enabled, the traffic analyzer gets directly attached to the port that is configured to receive a copy of every packet that host A sends. This port is called a traffic mirroring port.

**Note**

- From Release 7.4.2, you can mirror incoming (Rx) and outgoing (Tx) traffic from the source ports to separate destinations on Cisco NCS routers. During a session, you can configure one destination port for incoming traffic and one for outgoing traffic.

Traffic Mirroring Types

The following types of traffic mirroring are supported:

- **Local traffic mirroring:** This is the most basic form of traffic mirroring. The network analyzer or sniffer is attached directly to the destination interface. In other words, all monitored ports are located on the same router as the destination port.
- **Remote traffic mirroring:** The network analyzer is reached through a GRE tunnel over an IP network.

**Note**

A copy of every packet includes the Layer 2 header if the ethernet keyword is configured. As this renders the mirrored packets unroutable, the end point of the GRE tunnel must be the network analyzer.

- **ACL-based traffic mirroring:** Traffic is mirrored based on the configuration of the interface ACL.

You can mirror traffic based on the definition of an interface access control list. When you are mirroring Layer 3 traffic, the ACL is configured using the **ipv4 access-list** or the **ipv6 access-list** command with the **capture** option. The **permit** and **deny** commands determine the behavior of regular traffic. The **capture** option designates the packet is to be mirrored to the destination port, and it is supported only on permit type of access control entries (ACEs).

**Note**

Prior to Release 6.5.1, ACL-based traffic mirroring required the use of UDK (User-Defined TCAM Key) with the **enable-capture** option so that the **capture** option can be configured in the ACL.

- **Encapsulated remote SPAN (ERSPAN):** ERSPAN enables generic routing encapsulation (GRE) for all captured traffic and allows it to be extended across Layer 3 domains.
- **SPAN to File:** 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.
- **File Mirroring:** 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.

ERSPAN

Encapsulated Remote Switched Port Analyzer (ERSPAN) transports mirrored traffic over an IP network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface.

ERSPAN involves mirroring traffic through a GRE tunnel to a remote site. For more information on configuring the GRE tunnel that is used as the destination for the monitor sessions, see the chapter *Configuring GRE Tunnels*.

Restrictions

Generic Restrictions

The following are the generic restriction(s) related to traffic mirroring:

- Partial mirroring and sampled mirroring are not supported.
- Sub-interface configured as source interface is not supported on SPAN.
- The destination bundle interfaces flap when:
 - both the mirror source and destination are bundle interfaces in LACP mode and
 - mirror packets next-hop is a router or a switch instead of a traffic analyzer.

This behavior is observed due to a mismatch of LACP packets on the next-hop bundle interface due to the mirroring of LACP packets on the source bundle interface.

- Both SPAN and ERSPAN features cannot be configured on a router simultaneously. Either SPAN or ERSPAN feature can be configured on the same router.

SPAN Restrictions

The following restrictions apply to SPAN:

- SPAN only supports port-level source interfaces.
- SPAN counters are not supported.
- SPAN over pseudowire is not supported on the NCS 540 Routers.

ERSPAN Restrictions

The following restrictions apply to ERSPAN:

- The value of ERSPAN session-ID is always zero. IOS XR Command for configuring ERPAN is not available.
- ERSPAN next-hop must have ARP resolved. Any other traffic or protocol will trigger ARP.
- ERSPAN cannot travel over MPLS.
 - Additional routers may encapsulate in MPLS.
- ERSPAN tunnel statistics is not supported.
- ERSPAN decapsulation is not supported.
- ERSPAN does not work if the GRE next hop is reachable over sub-interface. For ERSPAN to work, the next hop must be reachable over the main interface.

SPAN-ACL Restrictions

The following restrictions apply to SPAN-ACL:

- SPAN-ACL is only supported in the Rx direction, that is, in the ingress direction v4 or v6 ACL.
- MPLS traffic cannot be captured with SPAN-ACL.
 - ACL for any MPLS traffic is not supported.

Configure Traffic Mirroring

These tasks describe how to configure traffic mirroring:

Configure Remote Traffic Mirroring

Procedure

Step 1 **configure**

Example:

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

Enters global configuration mode.

Step 2 **monitor-session *session-name***

Example:

```
RP/0/RP0/CPU0:router(config)# monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#
```

Defines a monitor session and enters monitor session configuration mode.

Step 3 **destination interface *tunnel-ip***

Example:

```
RP/0/RP0/CPU0:router(config-mon)# destination interface tunnelip3
```

Specifies the destination subinterface to which traffic is replicated.

Step 4 **exit****Example:**

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

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

Step 5 **interface** *type number***Example:**

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

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 6 **monitor-session** *session-name* **ethernet direction rx-onlyport-only****Example:**

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

Specifies the monitor session to be used on this interface. Use the **direction** keyword to specify that only ingress or egress traffic is mirrored.

Step 7 **end** or **commit****Example:**

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

or

```
RP/0/RP0/CPU0:router(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.

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:router# show monitor-session
```

Displays information about the traffic mirroring session.

Example

This example shows the basic configuration for traffic mirroring with physical interfaces.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session ms1
RP/0/RP0/CPU0:router(config-mon)# destination interface HundredGigE0/2/0/15
RP/0/RP0/CPU0:router(config-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE0/2/0/19
RP/0/RP0/CPU0:router(config-if)# monitor-session ms1 ethernet direction rx-only port-level
RP/0/RP0/CPU0:router(config-if)# commit
```

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

```
RP/0/RSP0/CPU0:router# show monitor-session status
Monitor-session cisco-rtpl
Destination interface HundredGigE 0/5/0/38
=====
Source Interface Dir Status
-----
TenGigE0/5/0/4 Both Operational
TenGigE0/5/0/17 Both Operational
RP/0/RSP0/CPU0:router# show monitor-session status detail
Monitor-session sess1
Destination interface is not configured
Source Interfaces
-----
TenGigE0/1/0/0
Direction: Both
ACL match: Disabled
Portion: Full packet
Status: Not operational (destination interface not known).
TenGigE0/1/0/1
Direction: Both
ACL match: Disabled
Portion: First 100 bytes

RP/0/RSP0/CPU0:router# show monitor-session status error
Monitor-session ms1
Destination interface TenGigE0/2/0/15 is not configured
=====
Source Interface Dir Status
-----
Monitor-session ms2
```

```

Destination interface is not configured
=====
Source Interface Dir Status
-----
RP/0/RP0/CPU0:router# show monitor-session test status
Monitor-session test (ipv4)
Destination Nexthop 255.254.254.4
=====
Source Interface Dir Status
-----
Gi0/0/0/2.2 Rx Not operational (source same as destination)
Gi0/0/0/2.3 Rx Not operational (Destination not active)
Gi0/0/0/2.4 Rx Operational
Gi0/0/0/4 Rx Error: see detailed output for explanation
RP/0/RP0/CPU0:router# show monitor-session test status error
Monitor-session test
Destination Nexthop ipv4 address 255.254.254.4
=====
Source Interface Status
-----
Gi0/0/0/4 < Error: FULL Error Details >

```

Attaching the Configurable Source Interface

Procedure

-
- Step 1** **configure**
- Example:**
- ```
RP/0/RP0/CPU0:router# configure
```
- Enters global configuration mode.
- Step 2**     **interface *type number***
- Example:**
- ```
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/1/0
```
- 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:router(config-if)# ipv4 access-group acl1 ingress
```
- Controls access to an interface.
- Step 4**     **monitor-session *session-name* ethernet direction rx-onlyport-level acl**
- Example:**

```
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only port-level
acl
RP/0/RP0/CPU0:router(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:router(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 step overrides any ACL that may be configured on the interface.

#### Step 6 **exit**

##### **Example:**

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

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

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

##### **Example:**

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

or

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

Displays information about the monitor session.

## Configuring UDF-Based ACL for Traffic Mirroring

**Procedure**

|               | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure</b><br><b>Example:</b><br><pre>RP/0/RP0/CPU0:router# configure</pre>                                                                                                                                                                                                                                                                                                                                                                                                                | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 2</b> | <b>udf <i>udf-name</i> header {inner   outer} {I2   I3   I4} offset <i>offset-in-bytes</i> length <i>length-in-bytes</i></b><br><b>Example:</b><br><pre>RP/0/RP0/CPU0:router(config)# udf udf3 header outer 14 offset 0 length 1 (config-mon)#</pre> <b>Example:</b><br><pre>RP/0/RP0/CPU0:router(config)# udf udf3 header inner 14 offset 10 length 2 (config-mon)#</pre> <b>Example:</b><br><pre>RP/0/RP0/CPU0:router(config)# udf udf3 header outer 14 offset 50 length 1 (config-mon)#</pre> | <p>Configures individual UDF definitions. You can specify the name of the UDF, the networking header from which offset, and the length of data to be extracted.</p> <p>The <b>inner</b> or <b>outer</b> keywords indicate the start of the offset from the unencapsulated Layer 3 or Layer 4 headers, or if there is an encapsulated packet, they indicate the start of offset from the inner L3/L4.</p> <p><b>Note</b> The maximum offset allowed, from the start of any header, is 63 bytes</p> <p>The <b>length</b> keyword specifies, in bytes, the length from the offset. The range is from 1 to 4.</p> |
| <b>Step 3</b> | <b>ipv4 access-list <i>acl-name</i></b><br><b>Example:</b><br><pre>RP/0/RP0/CPU0:router(config)# ipv4 access-list acl1</pre>                                                                                                                                                                                                                                                                                                                                                                     | Creates ACL and enters IP ACL configuration mode. The length of the <i>acl-name</i> argument can be up to 64 characters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 4</b> | <b>permit <i>regular-ace-match-criteria</i> udf <i>udf-name1</i> <i>value1</i> ... <i>udf-name8</i> <i>value8</i></b><br><b>Example:</b><br><pre>RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any udf udf1 0x1234</pre>                                                                                                                                                                                                                                                             | Configures ACL with UDF match.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

|               | Command or Action                                                                                                                                  | Purpose                                                                   |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|               | <pre>0xffff udf3 0x56 0xff capture RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 permit ipv4 any any dscp af11 udf udf5 0x22 0x22 capture</pre>        |                                                                           |
| <b>Step 5</b> | <p><b>exit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-ipv4-acl)# exit</pre>                                                   | Exits IP ACL configuration mode and returns to global configuration mode. |
| <b>Step 6</b> | <p><b>interface type number</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/1/0</pre>                  | Configures interface and enters interface configuration mode.             |
| <b>Step 7</b> | <p><b>ipv4 access-group acl-name ingress</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-if)# ipv4 access-group acl1 ingress</pre> | Applies access list to an interface.                                      |
| <b>Step 8</b> | <p><b>commit</b></p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-if)# commit</pre>                                                     | Applies access list to an interface.                                      |

## Additional Information on Traffic Mirroring

### Traffic Mirroring Terminology

- Ingress Traffic — Traffic that comes into the router.
- Egress Traffic — Traffic that goes out of the router.
- Source (SPAN) interface — An interface that is monitored using the SPAN feature.
- Source port—A port that is monitored 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 SPAN configurations consisting of a single destination and, potentially, one or many source interfaces.

## Characteristics of Source Port

A source port, also called a monitored port, is a routed port that you monitor for network traffic analysis. In a single traffic mirroring session, you can monitor source port traffic. The Cisco NCS540 Series router support a maximum of up to 800 source ports.

A source port has these characteristics:

- It can be any data port type, such as Bundle Interface, 100 Gigabit Ethernet, or 10 Gigabit Ethernet.



---

**Note**

- Bridge group virtual interfaces (BVI) are not supported.
  - Bundle members cannot be used as source ports.
- 

- Each source port can be monitored in only one traffic mirroring session.
- When a port is used as a source port, the same port cannot be used as a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor local traffic mirroring. Remote traffic mirroring is supported both in the ingress and egress directions. For bundles, the monitored direction applies to all physical ports in the group.

## Characteristics of 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 are 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 router can have a maximum of four monitor sessions. However, both SPAN and CFM share common mirror profiles. If you configure SPAN and CFM together on the router, the maximum number of monitor sessions may reduce to two.
- A single monitor session can have only one destination port.
- A single destination port can belong to only one monitor session.
- A monitor session can have a maximum of 800 source ports, as long as the maximum number of source ports from all monitoring sessions does not exceed 800.

## Characteristics of Destination Port

Each session must have a destination port that receives a copy of the traffic from the source ports.

A destination port has these characteristics:

- A destination port must reside on the same router as the source port for local traffic mirroring. For remote mirroring, the destination is always a GRE tunnel.

- A destination port for local mirroring can be any Ethernet physical port, EFP, GRE tunnel interface, or bundle interface. It can be a Layer 2 or Layer 3 transport interface.




---

**Note** Bundle members cannot be used as destination ports.

---

- A destination port on router cannot be a VLAN subinterface.
- At any one time, a destination port can participate in only one traffic mirroring session. A destination port in one traffic mirroring session cannot be a destination port for a second traffic mirroring session. In other words, no two monitor sessions can have the same destination port.
- A destination port cannot also be a source port.

## Traffic Mirroring Configuration Examples

This section contains examples of how to configure traffic mirroring:

### Traffic Mirroring with Physical Interfaces (Local): Example

This example shows the basic configuration for traffic mirroring with physical interfaces.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session ms1
RP/0/RP0/CPU0:router(config-mon)# destination interface HundredGigE0/0/1/0
RP/0/RP0/CPU0:router(config-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-if)# monitor-session ms1 ethernet direction rx-only port-level
RP/0/RP0/CPU0:router(config-if)# commit
```

### 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 HundredGigE 0/0/1/0
=====
Source Interface Dir Status

TenGigE0/0/0/4 Both Operational
TenGigE0/0/0/17 Both Operational

RP/0/RSP0/CPU0:router# show monitor-session status detail

Monitor-session sess1
Destination interface is not configured
Source Interfaces

```

```
TenGigE0/0/0/0
 Direction: Both
 ACL match: Disabled
 Portion: Full packet
 Status: Not operational (destination interface not known).
TenGigE0/0/0/1
 Direction: Both
 ACL match: Disabled
 Portion: First 100 bytes
```

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

```
Monitor-session ms1
Destination interface TenGigE0/0/0/15 is not configured
```

```
=====
Source Interface Dir Status

```

```
Monitor-session ms2
Destination interface is not configured
```

```
=====
Source Interface Dir Status

```

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

```
Monitor-session test (ipv4)
```

```
Destination Nexthop 255.254.254.4
```

```
=====
Source Interface Dir Status

```

```
Gi0/0/0/2.2 Rx Not operational (source same as destination)
Gi0/0/0/2.3 Rx Not operational (Destination not active)
Gi0/0/0/2.4 Rx Operational
Gi0/0/0/4 Rx Error: see detailed output for explanation
```

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

```
Monitor-session test
Destination Nexthop ipv4 address 255.254.254.4
```

```
=====
Source Interface Status

```

```
Gi0/0/0/4 < Error: FULL Error Details >
```

## 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:

```
show monitor-session status
Monitor-session 5
rx destination interface tunnel-ip5
tx destination is not specified
=====
Source Interface Dir Status

```

```
Te0/0/0/23 (port) Rx Operational
```

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. Review the <b>show</b> command output and ensure that a session with a correct name has been configured.                                                                                                                                                            |
| Destination interface <code>&lt;intf&gt;</code> ( <code>&lt;down-state&gt;</code> ) | The destination interface is not in Up state in the Interface Manager. You can verify the state using the <b>show interfaces</b> command. Check the configuration to determine 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. 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 <b>show</b> command output to ensure that a session with the right name has been configured.                                                                                                                              |
| Not operational (destination 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. 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 resolutions.                                                                                                                                   |
| Not operational (source state <code>&lt;down-state&gt;</code> ) | The source interface is not in the Up state. You can verify the state using the <b>show interfaces</b> 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 <b>show monitor-session status detail</b> command to display more information.                                                                                                                                                      |

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

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

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

```

TenGigE0/0/0/1
 Direction: Both
 ACL match: Disabled
 Portion: Full packet
 Status: Not operational (destination interface not known)
TenGigE0/0/0/2
 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 TenGigE 0/0/0/0
 Source Interfaces

TenGigE 0/0/0/1.100:
 Direction: Both
 Status: Operating
TenGigE 0/0/0/2.200:
 Direction: Tx
 Status: Error: <blah>

Monitor session bar
 No destination configured
 Source Interfaces

TenGigE 0/0/0/3.100:
 Direction: Rx
 Status: Not operational(no destination)

```

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 process all
RP/0/RP0/CPU0:router# debug monitor-session process ea
RP/0/RP0/CPU0:router# debug monitor-session process ma
RP/0/RP0/CPU0:router# show monitor-session process mgr

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
```

## Verifying UDF-based ACL

Use the **show monitor-session status detail** command to verify the configuration of UDF on ACL.

```
RP/0/RP0/CPU0:leaf1# show monitor-session 1 status detail

Fri May 12 19:40:39.429 UTC
Monitor-session 1
 Destination interface tunnel-ip3
 Source Interfaces

 TenGigE0/0/0/15
 Direction: Rx-only
 Port level: True
 ACL match: Enabled
 Portion: Full packet
 Interval: Mirror all packets
 Status: Not operational (destination not active)
```

## Introduction to ERSPAN Egress Rate Limit

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

ERSAPN 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 or the egress traffic. With rate limiting, you can limit the amount of egress traffic to a specific rate, which prevents the network and remote ERSPAN destination traffic overloading. Be informed, if the egress 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.
- The Discard Class (0 through 2):
  - The default is 0.
  - The discard class configuration is used in WRED.



### Benefits

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

## Topology

Figure 2: Topology for ERSPAN Egress 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 egress rate limit feature is applied on the router egress 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 Egress Rate Limit

Use the following steps to configure ERSPAN egress 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

```

!! Policy-map to be used with the ERSPAN Destination (egress interface)
!! Traffic class is set to 5. For packets in this class, apply shaping
!! as well as WRED.
class-map match-any TC5
 match traffic-class 5
end-class-map

```

```

!
policy-map shape-foo
class TC5
 random-detect discard-class 0 10000 bytes 40000 bytes
 random-detect discard-class 1 40000 bytes 80000 bytes
 random-detect discard-class 2 80000 bytes 200000 bytes
 shape average percent 15
!
class class-default
!
end-policy-map
!
!!GRE Tunnel Interface
interface Loopback49
 ipv4 address 49.49.49.49 255.255.255.255
!
interface tunnel-ip100
 ipv4 address 130.100.1.1 255.255.255.0
 tunnel mode gre ipv4
 tunnel source 49.49.49.49
 tunnel destination 10.8.1.2
!
!!ERSPAN Monitor Session with GRE tunnel as the Destination Interface, and with QoS
configuration
monitor-session FOO ethernet
 destination interface tunnel-ip100
 traffic-class 5
 discard-class 1
!
!!ERSPAN Source Interface
interface TenGigE0/6/0/4/0
 description connected to TGEN 9/5
 ipv4 address 10.4.90.1 255.255.255.0
 monitor-session FOO ethernet port-level
!
!
!!ERSPAN Destination ip-tunnel100's underlying interface, with egress policy-map shape-foo
attached
interface TenGigE0/6/0/9/0
 service-policy output shape-foo
 ipv4 address 10.8.1.1 255.255.255.0

```

## 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
 Portion: Full packet
 Interval: Mirror all packets
 Status: Operational
RP/0/RP0/CPU0:ios#
show monitor-session <sess-id> status internal

RP/0/RP0/CPU0:ios#show monitor-session FOO status internal
Wed May 2 15:13:06.063 UTC
Information from SPAN Manager and MA on all nodes:
Monitor-session FOO (ID 0x00000001) (Ethernet)

```

```

SPAN Mgr: Destination interface tunnel-ip100 (0x0800001c)
Last error: Success
Tunnel data:
 Mode: GREoIPv4
 Source IP: 49.49.49.49
 Dest IP: 10.8.1.2
 VRF:
 ToS: 0 (copied)
 TTL: 255
 DFbit: Not set
0/6/CPU0: Destination interface tunnel-ip100 (0x0800001c)
Tunnel data:
 Mode: GREoIPv4
 Source IP: 49.49.49.49
 Dest IP: 10.8.1.2
 VRF:
 ToS: 0 (copied)
 TTL: 255
 DFbit: Not set

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/6/CPU0: Name 'FOO', destination interface tunnel-ip100 (0x0800001c)
Platform, 0/6/CPU0:

 Dest Port: 0xe7d

ERSPAN Encap:
 Tunnel ID: 0x4001380b
 ERSPAN Tunnel ID: 0x4001380c
 IP-NH Grp key: 0x3140000cc5
 IP-NH hdl: 0x308a5fa5e0
 IP-NH IFH: 0x30002a0
 IP-NH IPAddr: 10.4.91.2

NPU MirrorRx MirrorTx
00 0x00000003 0x00000004
01 0x00000003 0x00000004
02 0x00000003 0x00000004
03 0x00000003 0x00000004
04 0x00000003 0x00000004
05 0x00000003 0x00000004
RP/0/RP0/CPU0:ios#

```

