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	Modification
Release 6.1.1	The local SPAN feature was introduced.
Release 6.1.31	Remote Traffic Monitoring feature was added.

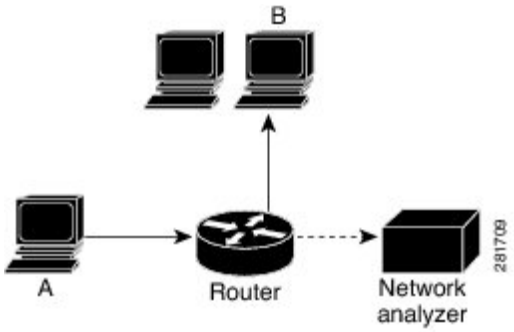
Introduction to Traffic Mirroring

Traffic mirroring, which is 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, and 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 is attached directly 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. The other sections of this document describe how you can fine tune this feature.

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 directly attached to the destination interface. In other words, all monitored ports are all 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 un-routable, the end 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 **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 is only supported on permit type of access control entries (ACE)s.

  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.

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 the 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 NCS 5500 Series Router can support a maximum of up to 800 source ports.

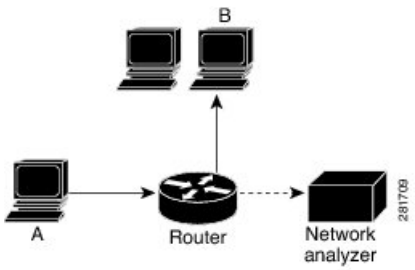
A source port has these characteristics:

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

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

- Each source port can be monitored in only one traffic mirroring session.
- When a port is used as a source port, the same cannot be used as a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor for 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.

Figure 2: Network Analysis on a Cisco NCS 5500 RouterWith 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 router can have a maximum of four monitor sessions.
- 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.

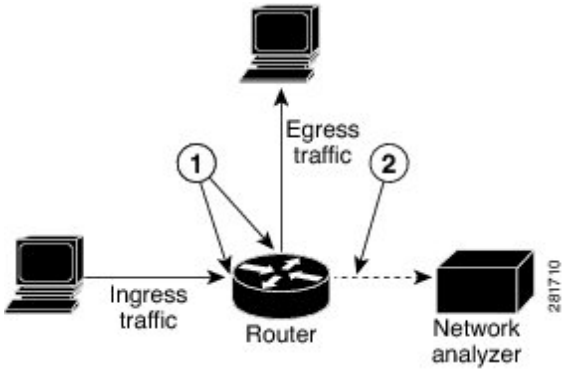
Characteristics of the Destination Port

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, and GRE tunnel interface, but not a bundle interface. It can be a Layer 2 or Layer 3 transport interface.
- A destination port on NCS5500 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.

Figure 3: Network Analysis on a Cisco NCS 5500 Series Router With Traffic Mirroring



In the above figure, the callouts indicate the following:

1. Source traffic mirroring ports (can be ingress or egress traffic ports).
2. Destination traffic mirroring port.

Restrictions

The following are the generic restriction(s):

- Partial mirroring and sampled mirroring are not supported.

The following general restrictions apply to traffic mirroring using ACLs:

- Traffic mirroring counters are not supported.
- ACL-based traffic mirroring is not supported with Layer 2 (ethernet-services) ACLs.
- Configure ACL(s) on the source interface or any interface on the same network processing unit as the source interface, to avoid default mirroring of traffic. If a Bundle interface is a source interface, configure the ACL(s) on any interface on the same network processing unit as all active bundle-members. Bundle members can be on multiple NPUs. Also ensure that the ACL(s) configured are of the same protocol type and direction as the SPAN configuration. For example, if you configure SPAN with ACL for IPv4 or IPv6, configure an ingress IPv4 or IPv6 ACL on that network processing unit respectively.

The following general restrictions apply to SPAN:

- SPAN only supports port-level source interfaces.

The following restrictions apply to ERSPAN and SPAN ACL:

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

How to Configure Traffic Mirroring

These tasks describe how to configure traffic mirroring:

Configuring Remote Traffic Mirroring

SUMMARY STEPS

1. **configure**
2. **monitor-session** *session-name*
3. **destination interface** *tunnel-ip*
4. **exit**
5. **interface** *type number*
6. **monitor-session** *session-name* **ethernet direction rx-onlyport-only**
7. **end** or **commit**
8. **show monitor-session [session-name] status** [**detail**] [**error**]

DETAILED STEPS

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 0/1/0/1
```

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.

Attaching the Configurable Source Interface

SUMMARY STEPS

1. **configure**
2. **interface** *type number*
3. **ipv4 access-group** *acl-name* {**ingress** | **egress**}
4. **monitor-session** *session-name* **ethernet direction rx-onlyport-level acl**
5. **acl**
6. **exit**
7. **end** or **commit**
8. **show monitor-session [session-name] status** [**detail**] [**error**]

DETAILED STEPS

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 0/1/0/1
```

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 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.

Traffic Mirroring Configuration Examples

This section contains examples of how to configure traffic mirroring:

Configuring ACLs for Traffic Mirroring

This section describes the configuration for creating ACLs for traffic mirroring.

Configuration

Use the following configuration to enable traffic mirroring with ACLs.

```
Router(config)# hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port
proto frag-bit enable-capture
Router(config)# commit
```

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 10.1.1.0 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
Router(config-ipv4-acl)# exit
Router(config)# commit

/* Validate the configuration */
Router(config)# show run
Thu May 17 11:17:49.968 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu May 17 11:17:47 2018 by user
…
hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port proto frag-bit
enable-capture

ipv4 access-list TM-ACL
 10 permit udp 10.1.1.0 0.0.0.255 eq 10 any capture
 20 permit udp 10.1.1.0 0.0.0.255 eq 20 any
!
…
```

You have successfully configured an IPv4 ACL for traffic mirroring.

Configuring UDF-Based ACL for Traffic Mirroring

SUMMARY STEPS

1. **configure**
2. **udf** *udf-name* **header** {**inner** | **outer**} {**l2** | **l3** | **l4**} **offset** *offset-in-bytes* **length** *length-in-bytes*

3. **hw-module profile tcam format access-list** {**ipv4 | ipv6**} [*acl-qualifiers*] [**udf1** *udf-name1 ...* **udf8** *udf-name8*] **enable-capture**
4. **ipv4 access-list** *acl-name*
5. **permit** *regular-ace-match-criteria* **udf** *udf-name1 value1 ... udf-name8 value8*
6. **exit**
7. **interface***type number*
8. **ipv4 access-group** *acl-name* **ingress**
9. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

Step 2	**udf** *udf-name* **header** {**inner	outer**} {**l2	l3	l4**} **offset** *offset-in-bytes* **length** *length-in-bytes*
				Example:
				`RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 0 length 1`
				`(config-mon)#`
				Example:
				`RP/0/RP0/CPU0:router(config)# udf udf3 header inner l4 10 length 2`
				`(config-mon)#`
				Example:
				`RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 50 length 1`
				`(config-mon)#`
					Configures the individual UDF definitions. You can specify the name of the UDF, the networking header to offset from, and the length of data to be extracted.
					The **inner** or **outer** keywords indicate the start of 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.
					Note The maximum offset allowed, from the start of any header is, 63 bytes
					The **length** keyword specifies the length from the offset, in bytes. Range is from 1 to 4.

Step 3	**hw-module profile tcam format access-list** {**ipv4	ipv6**} [*acl-qualifiers*] [**udf1** *udf-name1 ...* **udf8** *udf-name8*] **enable-capture** **Example:** `RP/0/RP0/CPU0:router(config)# hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port dst-port proto tcp-flags packet-length frag-bit udf1 udf-test1 udf2 udf-test2 enable-capture`	Adds the user-defined fields to the ACL key definition that is sent to the hardware. **Note** A reload of the line card is required for the new TCAM profile to take effect.
Step 4	**ipv4 access-list** *acl-name* **Example:**	Creates the ACL and enters the IP ACL configuration mode. The length of *acl-name* argument can be up to 64 characters.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config))# ipv4 access-list acl1`	

Step 5	**permit** *regular-ace-match-criteria* **udf** *udf-name1 value1 ... udf-name8 value8*
	Example:
	`RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any udf udf1 0x1234 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`
		Configures an ACL with UDF match.

Step 6	**exit** **Example:** `RP/0/RP0/CPU0:router(config-ipv4-acl)# exit`	Exits IP ACL configuration mode and returns to global configuration mode.
Step 7	**interface***type number* **Example:** `RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/2/0/2`	Configures an interface and enters interface configuration mode.
Step 8	**ipv4 access-group** *acl-name* **ingress** **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 access-group acl1 ingress`	Applies the access list to an interface.
Step 9	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Applies the access list to an interface.

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)
```

Traffic Mirroring with Physical Interfaces (Local): Example

This example shows a 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 port-level direction rx-only
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/RSP0/CPU0:router# show monitor-session status

Monitor-session cisco-rtp1
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 >
```

Monitoring Traffic Mirroring on a Layer 2 Interface

This section describes the configuration for monitoring traffic on a Layer 2 interface.

Configuration

To monitor traffic mirroring on a Layer 2 interface, configure the monitor under `l2transport` sub-config of the interface:

```
RP/0/RP0/CPU0:router(config)# interface TenGigE0/0/0/42
RP/0/RP0/CPU0:router(config-if)# l2transport
RP/0/RP0/CPU0:router(config-if-l2)# monitor-session EASTON ethernet port-level
```

Verification

```
RP/0/RP0/CPU0:router# show monitor-session status
Thu Aug 29 21:42:22.829 UTC
Monitor-session EASTON
Destination interface TenGigE0/0/0/20
=============================================
Source Interface      Dir   Status
--------------------- ----  -----------------
Te0/0/0/42 (port)     Both  Operational
```

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 ms1
```

```
<session status>
================================================================================


 Interface      Dir   Status
---------------------- ----  --------------------------------------------------
Gi0/1/0/0.10          Both  <Source interface status>
Gi0/1/0/0.11          Rx    <Source interface status>
Gi0/1/0/0.12          Tx    <Source interface status>
Gi0/2/0/0 (port)      Rx    <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> (<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 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 *Session status* error messages for suggested resolution.

Source Interface Status	Explanation
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/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 0/1/0/0.100:
  Direction: Both
  Status:  Operating
 TenGigE 0/2/0/0.200:
  Direction: Tx
  Status:  Error: <blah>

Monitor session bar
 No destination configured
 Source Interfaces
 -----------------
 TenGigE 0/3/0/0.100:
  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 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 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
```

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.

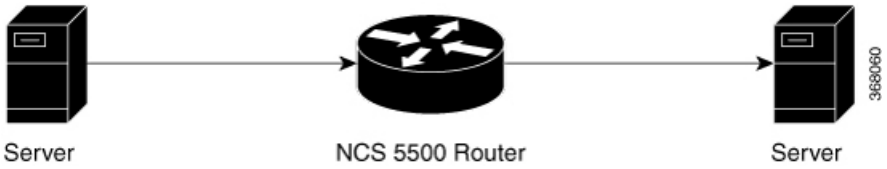
- The discard class configuration is used in WRED.

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 4: 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-tunnel00'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#
```