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). You can then pass this traffic to a destination port on the same router.

Feature Release History

Release	Modification
Release 7.2.1	File Mirroring was introduced.
Release 7.5.3	ERSPAN Traffic to a Destination Tunnel in a Non-Default VRF was introduced.
Release 7.5.4	• Multiple SPAN ACL Sessions in a Single Interface was introduced. • Monitor Multiple SPAN ACL and Security ACL Sessions was introduced. • SPAN Using 7-Tuples ACL was introduced. • DSCP Marking on Egress GRE Tunnel in ERSPAN was introduced. • DSCP Bitmask to filter Ingress SPAN was introduced. • Mirroring Forward-Drop Packets was introduced.
Release 7.6.1	VLAN Sub-interface as Ingress or Egress Source for Traffic Mirroring was introduced.

- Introduction to Traffic Mirroring, on page 2
- SPAN Types, Supported Features, and Configurations, on page 7
- Troubleshoot Traffic Mirroring, on page 35

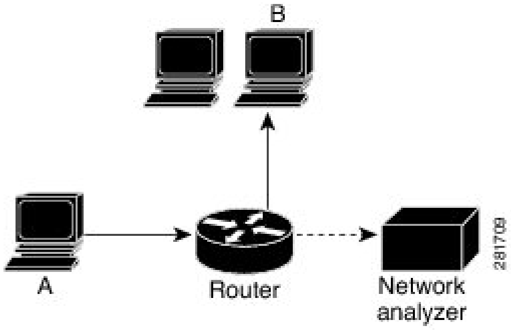
Introduction to Traffic Mirroring

Traffic mirroring, also referred to as 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 on a router. You can then mirror this traffic to a remote destination or 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 devices. 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 or network analyzer to the router and capture the ethernet traffic that is sent by host A to host B.

Figure 1: Traffic Mirroring Operation



Traffic Mirroring Terminology

- Ingress Traffic — Traffic that comes into the router.
- Egress Traffic — Traffic that goes out of the router.
- 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 ports.

Traffic Mirroring Types

These are the supported traffic mirroring types.

- Local SPAN
- Remote SPAN
- SPAN on Subinterfaces
- ACL-based SPAN

- ERSPAN
- Forward-Drop Packets Mirroring
- File Mirroring

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 routers 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 physical port, or 10 Gigabit Ethernet physical port.
- 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 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. This maximum limit is applicable only when 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 or file that receives a copy of the traffic from the source ports.

A destination port has these characteristics:

- A destination port cannot be a source port.
- 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.

- For remote mirroring, the destination is 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.
- A destination port on router cannot be a VLAN subinterface.
- At any 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.

Supported Scale

- Prior to Cisco IOS XR Software Release 7.8.1, a single router could support up to four monitor sessions. However, configuring SPAN and CFM on the router reduced the maximum number of monitor sessions to two, as both shared the mirror profiles.
- Starting Cisco IOS XR Software Release 7.8.1, up to three monitor sessions on are supported on the NCS 5500 router. But, if you configure SPAN and CFM on the router, the maximum number of monitor sessions decreases to one, as both functions use the same mirror profiles.

Restrictions

Generic Restrictions

The following are the generic restrictions 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 the Link Aggregation Control Protocol (LACP) mode.
  - 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.
- Subinterface with only one VLAN is supported as source for traffic mirroring.
- Bridge group virtual interfaces (BVIs) are not supported as source ports or destination ports.
- Bundle members cannot be used as destination ports.
- Fragmentation of mirror copies is not handled by SPAN when SPAN destination MTU is less than the packet size. Existing behaviour if the MTU of destination interface is less than the packet size is as below:

Platforms	Rx SPAN	Tx SPAN
NCS 5500	Mirror copies are not fragmented. Receives whole packets as mirror copies.	Mirror copies are fragmented.

You can configure the SPAN destination with an MTU which is greater than the packet size.

- Until Cisco IOS XR Software Release 7.6.1, SPAN only supports port-level source interfaces.
- SPAN counters are not supported.

VLAN Sub-interface as Source Restrictions

The following restrictions apply to VLAN sub-interface as source for traffic mirroring on NCS 540 routers from Cisco IOS XR Release 7.6.1:

- Supports a maximum of 24 reception and transmission sessions together for mirroring. This restriction is applicable for sub-intefaces and ports as source.
- When the port is in Egress Traffic Management (ETM) mode, the outbound or transmission mirroring is possible only on the sub-interface for which outbound traffic mirroring is configured.
- Transmission mirroring is applicable on ETM mode only. Reception mirroring is applicable on both the ETM and non-ETM modes.

SPAN Filtering on VLAN Interfaces Restrictions

These restrictions apply to SPAN filtering on Layer 2 and Layer 3 interfaces:

- On a main interface, if **span-acl** isn't configured and only **span** is configured, then the router performs only L2-L2 SPAN port filtering if **hw-module profile span-filter l2-rx-enable** command is enabled.
- Other Layer 2 point-to-point services such as Xconnect, VPWS, EVPN, and VPLS (PW) aren't supported.

ACL-based SPAN Restrictions

The following restrictions apply to SPAN-ACL:

Table 1: SPAN-ACL Support

Platforms	Rx Direction	Tx Direction
NCS 5500	Supported at the port level, that is, in the ingress direction for IPv4 or IPv6 ACLs.	Not supported.

- Multi-SPAN ACL is supported in the Rx direction in Cisco IOS XR Release 7.5.4 and from Cisco IOS XR Release 7.10.1.
- Multi-SPAN ACL sessions can be used only with SPAN Using 7-Tuples ACL.
- MPLS traffic cannot be captured with SPAN-ACL.
    - ACL for any MPLS traffic is not supported.

- Traffic mirroring counters are not supported.
- ACL-based traffic mirroring is not supported with Layer 2 (ethernet-services) ACLs.
- Main interface as span source interface and ACL with the **capture** keyword on same main interface's sub-interface are not supported.
- If a SPAN session with the **acl** keyword is applied on an interface with no ACL rule attached to that interface, SPAN happens without any filtering.
- Configure one or more ACLs on the source interface to avoid default mirroring of traffic. If a Bundle interface is a source interface, configure the ACL on the bundle interface (not bundle members). Also, ensure that the ACL configured is a UDK (with capture field) and 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 UDK (with capture) or IPv6 UDK (with capture) on that network processing unit respectively.
- Configure one or more ACLs 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 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 ACLs 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.

ERSPAN Restrictions

This section provides the restrictions that apply to ERSPAN and multiple ERSPAN sessions.

The following restrictions apply to ERSPAN:

- The value of ERSPAN session-ID is always zero. IOS XR command for configuring ERSPAN is not available.
- ERSPAN next-hop must have ARP resolved. Any other traffic or protocol will trigger ARP.
- ERSPAN packets with outgoing interface having MPLS encapsulation are not supported.
  - Additional routers may encapsulate in MPLS.
- ERSPAN sessions can be created only on physical interfaces. The sessions cannot be created on sub-interfaces.
- ERSPAN supports a maximum of three sessions.
- 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.
- ERSPAN decapsulation is not supported. Tunnel destination should be network analyzer.

Multiple ERSPAN Restrictions

- All sessions under the source port should have SPAN access control list (ACL) enabled.

- A few sessions with SPAN ACL and a few without SPAN ACLs in the same source interface are not supported.
- No two sessions should have the same ACL in the same source interface. Each session should have a different ACL.
- Multiple sessions without ACL in the same interface are not supported.
- Multi-SPAN ACL does not support the **Deny** action.
- One SPAN session with the keyword ACL (use security acl as the keyword) and other SPAN sessions with the keyword SPAN ACL are not supported.
- At a time, you can make only one mirror copy of a packet.
- Capturing keywords is not required.
- Multiple sessions under the same interface cannot have a combination of directions. Only RX is supported.

File Mirroring Restrictions

The following restrictions 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.

Forward-Drop Packets Mirroring Restrictions

These are some restrictions for Forward-Drop packets mirroring:

- Only one global forward-drop session can be configured on a router.
- When traffic-class is configured under monitor-session for forward-drop, the type of service (ToS) byte of the outgoing ERSPAN packet is overwritten with the configured traffic-class value.
- In-band traffic destined to router management interface cannot be captured using this functionality.

SPAN Types, Supported Features, and Configurations

Local SPAN

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 SPAN

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** *subinterface*

Example:

```
RP/0/RP0/CPU0:router(config-mon)# destination interface TenGigE 0/2/0/4.1
```

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 port-level
RP/0/RP0/CPU0:router(config-if)# 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 direction rx-only port-level
RP/0/RP0/CPU0:router(config-if)# 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 direction tx-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-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 >
```

SPAN on Subinterfaces

VLAN Subinterface as Ingress or Egress Source for Traffic Mirroring

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
VLAN Subinterface as Ingress or Egress Source for Traffic Mirroring	Release 7.6.1	You can now configure the VLAN subinterface as an egress or ingress source for traffic mirroring. This feature enables the monitoring of traffic mirrored on either egress or ingress or both directions. You could configure mirror functionality only at the main interface level in earlier releases.

VLAN subinterface provides the flexibility to monitor ingress or egress, or both ingress/egress traffic from all the active subinterfaces of the source VLAN. The active subinterfaces in the source VLAN are considered as source subinterfaces. When subinterfaces are added or removed from the source VLAN, the corresponding traffic is added or removed from the monitoring sources.

VLAN Interface as Ingress Source for Traffic Mirroring

Configuration Example

```
Router# configure
Router(config)# monitor-session mon1 ethernet
Router(config-mon)# destination interface tunnel-ip 3
Router(config-mon)# exit
Router(config)# interface HundredGigE 0/1/0/1.10
Router(config-subif)#
Router(config-if-mon)# commit
```

Running Configuration

```
Router# show run monitor-session mon1
monitor-session mon1 ethernet
 destination interface tunnel-ip3
!
```

```
Router# show run interface HundredGigE 0/1/0/1.10
 interface HundredGigE0/1/0/1.10
  encapsulation dot1q 10
   ipv4 address 101.1.2.1 255.255.255.252
    monitor-session mon1 ethernet
     !
    !
   !
```

Verification

```
Router# show monitor-session status
Monitor-session mon1
Destination interface tunnel-ip3
```

```
================================================================================
Source Interface Dir Status
--------------------- ---- ----------------------------------------------------
HundredGigE 0/1/0/1.10  Both Operational
```

VLAN Interface as Egress Source for Traffic Mirroring

Configuration Example

```
Router# configure
Router(config)# controller optics 0/0/0/1
Router(config-Optics)# mode etm
Router(config-Optics)# exit
Router(config)# interface HundredGigE 0/1/0/1.10
Router(config-subif)#
Router(config-if-mon)# commit
```

Running Configuration

```
Router# show run monitor-session mon1
monitor-session mon1 ethernet
 destination interface tunnel-ip3
!


Router# show run interface HundredGigE 0/1/0/1.10
interface HundredGigE0/1/0/1.10
 encapsulation dot1q 20
  ipv4 address 102.1.2.1 255.255.255.252
   monitor-session mon1 ethernet
    !
   !
  !
```

Verification

```
Router# show monitor-session status
Monitor-session mon1
Destination interface tunnel-ip3
================================================================================
Source Interface Dir Status
--------------------- ---- ----------------------------------------------------
HundredGigE 0/1/0/1.10  Both Operational
```

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
```

ACL-based SPAN

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 mirror 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 if the packets in the traffic are permitted or denied. 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.

Configuring Security ACLs for Traffic Mirroring

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

In 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're mirroring Layer 3 or Layer 2 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 the regular traffic.

Configure an IPv4 ACL for Traffic Mirroring

Use the following steps 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
…
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 Security ACL for Traffic Mirroring

Before you begin

This section describes the configuration steps for UDF-based security ACLs for traffic mirroring.

Procedure

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 offset 0 length 1
(config-mon)#
```

Example:

```
RP/0/RP0/CPU0:router(config)# udf udf3 header inner l4 offset 10 length 2
(config-mon)#
```

Example:

```
RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 offset 50 length 1
(config-mon)#
```

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.

The **inner** or **outer** 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.

Note The maximum offset allowed, from the start of any header, is 63 bytes

The **length** keyword specifies, in bytes, the length from the offset. The range is from 1 to 4.

Step 3 **ipv4 access-list** *acl-name*

Example:

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

Creates ACL and enters IP ACL configuration mode. The length of the *acl-name* argument can be up to 64 characters.

Step 4 **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 ACL with UDF match.

Step 5 **exit**

Example:

```
RP/0/RP0/CPU0:router(config-ipv4-acl)# exit
```

Exits IP ACL configuration mode and returns to global configuration mode.

Step 6 **interface***type number*

Example:

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

Configures interface and enters interface configuration mode.

Step 7 **ipv4 access-group** *acl-name* **ingress**

Example:

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

Applies access list to an interface.

Step 8 **commit**

Example:

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

Applies access list to an interface.

Verifying UDF-based Security ACL

Use the **show monitor-session status detail** command to verify the configuration of UDF on security 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)
```

DSCP Bitmask to Filter Ingress SPAN Traffic

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
DSCP Bitmask to Filter Ingress SPAN Traffic	Release 7.5.4	You can now mirror multiple traffic flows for matched Differentiated Service Code Point (DSCP) value of IP header on the SPAN. The matched DSCP value is based on the DSCP value and the bitmask configured in Access Control List (ACL) rule. Earlier, you could monitor single traffic flow by setting the RFC 4594 defined DSCP values in the IP header. This feature introduces the following changes: • **CLI:** 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 SPAN to mirror specific traffic flows.

Without ACL rule, SPAN mirrors all the traffic on the incoming port. When ACL is configured with DSCP and DSCP mask on the SPAN, SPAN 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 SPAN 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 SPAN 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 SPAN*/
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 SPAN 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 ERPSAN*/
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 Using 7-Tuples ACL

Table 4: Feature History Table

Feature Name	Release Information	Description
SPAN Using 7-Tuples ACL	Release 7.5.4	With this release, you can perform packet capturing with 7-tuple access control lists (ACL). This capability allows you to define seven specific attributes in the ACL and apply it to an interface using the **monitor-session** command. The 7-tuple parameters include source and destination IP addresses, source and destination port numbers, and so on. When the 7-tuples are configured in the ACL, only the matching packets are captured and mirrored. The administrators can examine the captured packets and identify issues such as network congestion and security threats. This analysis helps in diagnosing and resolving network problems, enhancing network performance, and ensuring robust security measures.

Packet capturing functionality enables the network administrators to capture and analyze packets that pass through a router. By defining the seven parameters in the ACL, known as the 7-tuples, data packets can be matched and captured. Only packets that satisfy any or all of the seven parameters are mirrored. The captured packets can be analyzed locally or can be saved and exported for offline analysis.

The following parameters can be included in a 7-tuple ACL:

- Source IP Address (`source ip prefix`)
- Destination IP Address (`dest ip prefx`)
- Protocol (`protocol`, for example, TCP, UDP)
- Differentiated services code point `DSCP`
- Source Port (`source port`)
- Destination Port (dest port)
- Multiple TCP flags

By leveraging this level of granularity, you can fine-tune the packet capturing process to focus on the data relevant to your monitoring objectives.

Configuration Example

You can define the ACL with the seven tuples and apply it to the interface. Use the following sample configuration:

```
RP/0/RP0/CPU0:ios#config
Tue Jul 23 08:35:18.506 UTC
RP/0/RP0/CPU0:ios(config)#ipv4 access-list v4-monitor-acl2
RP/0/RP0/CPU0:ios(config-ipv4-acl)#80 permit tcp 80.1.1.0 0.0.0.255 eq www 30.30.30.0
0.0.0.255 eq www fin dscp af11
RP/0/RP0/CPU0:ios(config-ipv4-acl)#commit
```

```
Tue Jul 23 08:37:05.265 UTC
RP/0/RP0/CPU0:ios(config-ipv4-acl)#exit
RP/0/RP0/CPU0:ios(config)#ipv6 access-list v6-monitor-acl2
RP/0/RP0/CPU0:ios(config-ipv6-acl)#80 permit tcp 8010::/64 eq www 3010::/64 eq www fin dscp
 af11
RP/0/RP0/CPU0:ios(config-ipv6-acl)#commit
Tue Jul 23 08:37:39.689 UTC
RP/0/RP0/CPU0:ios(config-ipv6-acl)#exit
RP/0/RP0/CPU0:ios(config)#exit
RP/0/RP0/CPU0:ios#
```

The following example shows ERSPAN with QoS Configuration:

```
Router#configure
/* GRE Tunnel Interface */
Router(config)#interface Loopback49
Router(config-if)#ipv4 address 172.16.0.1 255.240.0.0
Router(config-if)#exit
Router(config)#interface tunnel-ip100
Router(config-if)#ipv4 address 192.168.0.1 255.255.0.0
Router(config-if)#tunnel mode gre ipv4
Router(config-if)#tunnel source 49.49.49.49
Router(config-if)#tunnel destination 10.0.0.2
Router(config-if)#exit
/* ERSPAN Monitor Session with GRE tunnel as the Destination Interface, and with QoS
configuration */
Router(config)#monitor-session FOO ethernet
Router(config-mon)#destination interface tunnel-ip100
Router(config-mon)#traffic-class 5
Router(config-mon)#discard-class 1
Router(config-mon)#exit
/* ERSPAN Source Interface */
Router(config)#interface TenGigE0/6/0/4/0
Router(config-if)#description connected to TGEN 9/5
Router(config-if)#ipv4 address 10.0.0.1 255.0.0.0
Router(config-if)#monitor-session FOO ethernet port-level
Router(config-if-mon)#acl ipv4 v4-monitor-acl2
Router(config-if-mon)#acl ipv6 v6-monitor-acl2
Router(config-if-mon)#exit
Router(config-if)#exit
/* ERSPAN Destination ip-tunnel00's underlying interface, with egress policy-map shape-foo
 attached */
Router(config)#interface TenGigE0/6/0/9/0
Router(config-if)#service-policy output shape-foo
Router(config-if)#ipv4 address 10.0.0.3 255.0.0.0
Router(config-if)#commit
```

Verification

Use **show monitor-session status** command to get the details of the monitor session.

```
Router# show monitor-session status
```

Displays information about the monitor session.

Multiple SPAN ACL Sessions in Single Interface

Table 5: Feature History Table

Feature Name	Release Information	Description
Multiple SPAN ACL Sessions in a Single Interface	Release 7.5.4	With this release, you can configure multiple SPAN ACL sessions under a single interface. A maximum of three sessions can be configured simultaneously. This feature, which is supported on layer 3 interfaces, helps you in monitoring traffic from different parts of your network simultaneously to see the network's overall performance. In addition, using this feature, you can get a better network visibility, more efficient use of network resources, and flexibility. You should specify the monitor sessions to be used on the interface. Use the **monitor-session** *session name* **ethernet direction rx-only port-level** command to specify that only the ingress traffic is mirrored. This feature is not supported on subinterfaces.

This feature allows you to configure multiple SPAN ACL sessions in the same source interface. The maximum number of sessions that are supported under an interface is three. The ACL is applicable only in the ingress direction (direction Rx). This configuration is supported only on Layer 3 interfaces.

To differentiate multiple SPAN sessions under the same source interface, span session ID is used. When a packet matches multiple entries at the router, priority attribute is used to choose the correct destination for the packet. When a single packet tries to match multiple SPAN sessions, you should configure correct priority fields to identify the correct destination. The ACL with the lowest priority is chosen.

For Cisco NCS 5500 routers, the merge group value is always 1, and the priority value can be of any value within the supported range of 1 to1000.

Multiple SPAN ACL sessions in a single interface help the administrators in the following ways:

- Monitor traffic from different parts of your network simultaneously to see the overall network performance.
- Isolate traffic from specific networks for troubleshooting network issues.
- Segment traffic for different purposes, such as security, compliance, or performance analysis.

Configure Multiple SPAN ACL Sessions

Configure Multiple SPAN ACL Sessions

Specify the monitor sessions to be used on the interface. Use the direction keyword to specify that only ingress traffic is mirrored. See the following example:

```
Router(config)#interface TenGigE0/0/0/26
Router(config-if)#monitor-session ses1 ethernet direction rx-only port-level
Router(config-if)#acl ipv4 acl1
!
Router(config-if)#monitor-session ses2 ethernet direction rx-only port-level
Router(config-if)#acl ipv4 acl2
!
Router(config-if)#monitor-session ses3 ethernet direction rx-only port-level
Router(config-if)#acl ipv4 acl3
!
Router(config-if)#monitor-session ses4 ethernet direction rx-only port-level
Router(config-if)#acl ipv4 acl4
!
!
```

Verify the Sessions

The following example shows the details of the monitor sessions.

```
Router##sh monitor-session status
Tue Mar 21 16:14:15.879 UTC
Monitor-session ses1
Destination interface TenGigE0/0/0/9
================================================================================
Source Interface      Dir   Status
--------------------- ----  ----------------------------------------------------
Te0/0/0/0 (port)      Rx    Operational

Monitor-session ses2
Destination interface TenGigE0/0/0/1
================================================================================
Source Interface      Dir   Status
--------------------- ----  ----------------------------------------------------
Te0/0/0/0 (port)      Rx    Operational

Monitor-session ses3
Destination interface TenGigE0/0/0/2
================================================================================
Source Interface      Dir   Status
--------------------- ----  ----------------------------------------------------
Te0/0/0/0 (port)      Rx    Operational

RP/0/RP0/CPU0:ios#
```

Configuring the Correct Priority

When one packet tries to match more than one SPAN session, the priority field helps in identifying the correct destination.

Note Merge group and priority fields are not mandatory. But if used, configure both fields.

```
Router(config)#interface tenGigE 0/0/0/24
Router(config-if)#monitor-session ses1 ethernet port-level
Router(config-if)#acl ipv4 acl1 merge-group 1 priority 30
```

To verify the traffic, use the following sample **show monitor-session** command:

```
Router#show monitor-session status detail
Tue Mar 21 16:15:02.741 UTC
Monitor-session ses1
  Destination interface TenGigE0/0/0/9
  Source Interfaces
  -----------------
  TenGigE0/0/0/0
    Direction:    Rx-only
    Port level:   True
    ACL match:    Disabled
    IPv4 ACL:     Enabled (acl1, merge-group: 1,priority: 1)
    IPv6 ACL:     Disabled
    Portion:      Full packet
    Interval:     Mirror all packets
    Mirror drops: Disabled
    Status:       Operational

Monitor-session ses2
  Destination interface TenGigE0/0/0/1
  Source Interfaces
  -----------------
  TenGigE0/0/0/0
    Direction:    Rx-only
    Port level:   True
    ACL match:    Disabled
    IPv4 ACL:     Enabled (acl2)
    IPv6 ACL:     Disabled
    Portion:      Full packet
    Interval:     Mirror all packets
    Mirror drops: Disabled
    Status:       Operational

Monitor-session ses3
  Destination interface TenGigE0/0/0/2
  Source Interfaces
  -----------------
  TenGigE0/0/0/0
    Direction:    Rx-only
    Port level:   True
    ACL match:    Disabled
    IPv4 ACL:     Enabled (acl3)
    IPv6 ACL:     Disabled
    Portion:      Full packet
    Interval:     Mirror all packets
    Mirror drops: Disabled
    Status:       Operational

Monitor-session ses4
  Destination interface TenGigE0/0/0/6
  Source Interfaces
  -----------------
  TenGigE0/0/0/0
    Direction:    Rx-only
    Port level:   True
    ACL match:    Disabled
    IPv4 ACL:     Enabled (acl4)
    IPv6 ACL:     Disabled
    Portion:      Full packet
    Interval:     Mirror all packets
    Mirror drops: Disabled
    Status:       Operational
Router#
```

Monitor Multiple SPAN ACL and Security ACL Sessions

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Monitor Multiple SPAN ACL and Security ACL Sessions	Release 7.5.4	With this feature, you can use SPAN and security ACLs together to monitor multiple SPAN ACL sessions under the same source interface. SPAN ACL helps you to distribute the mirrored traffic over different destination interfaces. Security ACL allows selective incoming traffic.

Starting Cisco IOS XR Software Release 7.5.4 you can monitor multiple ERSPAN sessions using GREv4 under the same source interface. Multiple SPAN ACL monitor sessions configured on an interface allow you to choose the destination interface for the mirrored traffic. For the configuration of monitor sessions, you can use SPAN and security ACLs together.

The SPAN and security ACLs apply only in the ingress traffic.

Configure Multiple SPAN ACL and Security ACL Monitor Sessions

This example shows how to attach the SPAN and security ACLs to configure multiple monitoring sessions.

Configuration example

Use the following configuration to attach SPAN and security ACLs for traffic mirroring.

```
Router# config
/*Perform the following configurations to attach the SPAN ACL to an interface*/
Router(config-if)#monitor-session always-on-v4 ethernet direction rx-only port-level
Router(config-if-mon)#acl ipv4 v4-monitor-acl1
Router(config-if-mon)#acl ipv6 v6-monitor-acl1
Router(config-if-mon)#exit
Router(config-if)#monitor-session on-demand-v4 ethernet direction rx-only port-level
Router(config-if-mon)#acl ipv4 v4-monitor-acl2
Router(config-if-mon)#acl ipv6 v6-monitor-acl2
Router(config-if-mon)#exit
/*Perform the following configurations to attach the security ACL to an interface*/
Router(config-if)#ipv4 access-group sec_aclv4 ingress
Router(config-if)#ipv6 access-group sec_aclv6 ingress
Router(config-if)#commit
```

Running configuration

```
Router(config)#show running-config interface
monitor-session always-on-v4 ethernet direction rx-only port-level
  acl ipv4 v4-monitor-acl2
  acl ipv6 v6-monitor-acl2
!
monitor-session on-demand-v4 ethernet direction rx-only port-level
  acl ipv4 v4-monitor-acl2
  acl ipv6 v6-monitor-acl2
!
```

```
ipv4 access-group sec_aclv4 ingress
ipv6 access-group sec_aclv6 ingress
!
!
```

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.

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.

Encapsulated Remote SPAN (ERSPAN) enables generic routing encapsulation (GRE) for all captured traffic and allows it to be extended across Layer 3 domains.

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*.

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.

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.

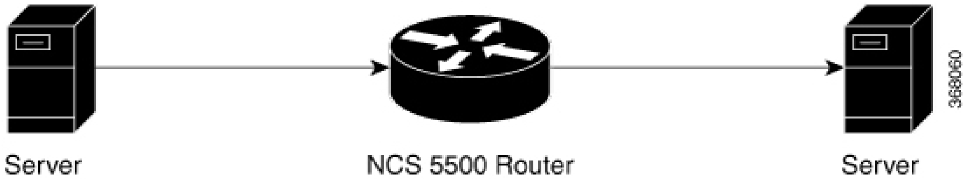
- Traffic Class (0 through 7)
  - Traffic class 0 has the lowest priority and 7 the highest.
  - The default traffic class is the same as that of the original traffic class.
- The Discard Class (0 through 2):
  - The default is 0.
  - The discard class configuration is used in WRED.

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-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#
```

ERSPAN Traffic to a Destination Tunnel in a Non-Default VRF

Table 7: Feature History Table

Feature Name	Release Information	Description
ERSPAN Traffic to a Destination Tunnel in a Non-Default VRF	Release 7.5.3	The tunnels are grouped under the VRFs and you can segregate the traffic towards a specific VRF domain. Encapsulated Remote Switched Port Analyzer (ERSPAN) now transports mirrored traffic through GRE tunnels with multiple VRFs, helping you design your network with multiple Layer 3 partitions. In earlier releases, ERSPAN transported mirrored traffic through GRE tunnels that belonged to only default VRF.

Here, the tunnel interface, where the traffic mirroring is destined, is now in a VRF.

The traffic coming out of the interfaces of a router do not have any grouping. By configuring a specific VRF, you can now identify the incoming traffic group.

Configuration

Use the following command to configure a specific VRF:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-ip 2
RP/0/RP0/CPU0:router(config)# tunnel vrf red
```

For more information on enabling the tunnel mode in GRE, see Configuring GRE Tunnels.

Configuration example

The following example shows a tunnel interface configured with endpoints in a non-default VRF (**vrf: red**):

```
Router#show run int tunnel-ip 2
Thu Feb  3 06:18:28.075 UTC
interface tunnel-ip2
 ipv4 address 102.1.1.100 255.255.255.0
 tunnel tos 32
 tunnel mode gre ipv4
 tunnel source 120.1.1.100
 tunnel vrf red
 tunnel destination 120.1.1.1
```

```
Router#show monitor-session status
Thu Feb  3 06:18:11.061 UTC
Monitor-session ERSPAN-2
Destination interface tunnel-ip2
===============================================================
Source Interface      Dir   Status
--------------------- ----  ----------------------------------------------------
Te0/0/0/5 (port)      Rx    Operational
```

Verification

The following CLI output shows how to verify, if the configured tunnel VRF is programmed in the session:

```
Router#show monitor-session ERSPAN-2 status internal
Thu Feb  3 06:19:50.014 UTC

Information from SPAN Manager and MA on all nodes:
Monitor-session ERSPAN-2 (ID 0x00000003) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip2 (0x20008024)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 120.1.1.100
            Dest IP: 120.1.1.1
            VRF: red
            VRF TBL ID: 0
            ToS: 32
            TTL: 255
            DFbit: Not set
```

DSCP Marking on Egress GRE Tunnel in ERSPAN

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
DSCP Marking on Egress GRE Tunnel in ERSPAN	Release 7.5.4	You can now set or modify Differentiated Service Code Point (DSCP) value on the ERSPAN GRE tunnel header. This feature allows you to control the QoS for your network's ERSPAN GRE tunnel traffic and eases the effort to control your customers' bandwidth across next-hop routers.

Starting Cisco IOS XR Release 7.5.4, you can set or modify DSCP marking on ERSPAN GRE tunnels. ERSPAN uses GRE encapsulation to route SPAN capture traffic.

Configure DSCP Marking on Egress GRE Tunnel in ERSPAN

Configuration Example

This example shows how you can configure DSCP Marking on Egress GRE tunnel in ERSPAN.

```
Router#configure terminal
Router(config)#interface tunnel-ip1
Router(config-if)#tunnel tos 96
Router(config-if)#tunnel mode gre ipv4
Router(config-if)#tunnel source 192.0.2.1
Router(config-if)#tunnel destination 192.0.2.254
```

Note You can configure DSCP value on both IPv4 and IPv6 headers.

Running Configuration

```
interface tunnel-ip1
 tunnel tos 96
 tunnel mode gre ipv4
 tunnel source 192.0.2.1
 tunnel destination 192.0.2.254
!
```

Verification

You can use the following commands to verify that tos value is configured:

```
Router#show run interface tunnel-ip 1
interface tunnel-ip1
 ipv4 address 192.0.2.0/24
 tunnel tos 96
 tunnel mode gre ipv4
 tunnel source 192.0.2.1
 tunnel vrf red
 tunnel destination 192.0.2.254
```

```
Router#show monitor-session ERSPAN-2 status internal

Information from SPAN Manager and MA on all nodes:
Monitor-session ERSPAN-2 (ID 0x00000003) (Ethernet)
SPAN Mgr: Destination interface tunnel-ip1 (0x20008024)
          Last error: Success
          Tunnel data:
            Mode: GREoIPv4
            Source IP: 192.0.2.1
            Dest IP: 192.0.2.254
            VRF: red
            VRF TBL ID: 0
            ToS: 96
            TTL: 255
            DFbit: Not set
```

File Mirroring

Prior to Cisco IOS XR Software Release , 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 , the 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.

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
```

Forward-Drop Packets Mirroring

In a network, packets are forwarded from one device to another until they reach their destination. However, in some cases, routers may drop packets during this forwarding process. These packets are known as forward-drop packets.

Packets can be dropped for several reasons such as congestion on the network, errors in the packet header or payload, blocking by firewall or access control lists (ACL), and so on. These forward-drop packets are typically discarded before they can reach their intended destination, and may have to be re-transmitted by the source device. This feature supports mirroring of these forward-drop packets at the ingress (Rx direction) to another destination. When a global forward-drop session is configured for the router, the forward-drop packets at the ingress are mirrored or copied to the configured destination. You can configure the mirror destination as an IPv4 GRE tunnel ID (for ERSPAN).

Mirror Forward-Drop Packets

Table 9: Feature History Table

Feature Name	Release Information	Description
Mirror Forward-Drop Packets	Release 7.5.4	Mirroring forward-drop packets feature copies or mirrors the packets that are dropped during the forwarding process at the router ingress to a configured destination. These mirrored packets can be captured and analyzed using network monitoring tools. The analysis of dropped packets helps you understand the types of traffic that are blocked, analyze potential security threats, troubleshoot, and optimize network performance. This feature introduces the following changes: • **CLI: forward-drop rx** • **YANG Data Model:**New XPath for Cisco-IOS-XR-um-monitor-session-cfg.yang (see GitHub, YANG Data Models Navigator)

Mirroring forward-drop packets to a suitable destination for analysis can help in the following:

- Network visibility: By mirroring and analyzing forward-drop packets, network administrators gain better visibility into the types of traffic that are blocked by the firewalls and access control lists (ACL).
- Threat detection: As the original dropped packet is forwarded without any change, it helps in identifying the source of potential security threats.
- Troubleshooting: Analyzing forward-drop packets helps in troubleshooting network issues that may be causing the packet drop. This helps in taking proactive measures to avoid escalation of the issue.

Configure Forward-Drop Mirroring

Perform the following tasks on the router to configure a global session for mirroring forward-drop packets:

1. Configure the tunnel mode.
2. Configure the tunnel source.
3. Configure the tunnel destination.
4. Configure a traffic mirroring session.
5. Associate a destination interface with the traffic mirroring session.
6. Run **forward-drop rx** command to start mirroring forward-drop packets.

This example shows how to configure a global traffic mirroring session for forward-drop packets.

```
Router(config)# interface tunnel-ip 2
Router(config-if)# tunnel mode gre ipv4
Router(config-if)# tunnel source 20.20.20.20
Router(config-if)# tunnel destination 192.1.1.3
Router(config-if)!
Router(config)# monitor-session mon2 ethernet
Router(config)#destination interface tunnel-ip2
Router(config)#forward-drop rx
Router(config)#!
```

Running Configuration

This section shows forward-drop running configuration.

```
RP/0/RSP0/CPU0:router#sh running-config
interface tunnel-ip 2
tunnel mode gre ipv4
tunnel source 20.20.20.20
tunnel destination 192.1.1.3
!
monitor-session mon2 ethernet
destination interface tunnel-ip2
forward-drop rx
!
```

Verification

Verify the forward-drop packets are mirrored using the **show monitor-session** command.

```
Router#show monitor-session mon2 status detail
Mon Aug 15 19:14:31.975 UTC
Monitor-session mon2
   Destination interface tunnel-ip2
   All forwarding drops:
      Direction: Rx
Source Interfaces
-----------------
```

Troubleshoot 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 command output and ensure that a session with a correct name configured.

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

The <Source interface status> can report these messages:

Source Interface Status	Explanation
Operational	Everything appears to be working correctly in traffic mirroring PI. follow up with the platform teams in the first instance, if mirroring operating as expected.
Not operational (Session is not configured globally)	The session does not exist in global configuration. Check the **show** command output to ensure that a session with the right name has b configured.
Not operational (destination not known)	The session exists, but it either does not have a destination interface s or the destination interface named for the session does not exist. For 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 inte traffic mirroring does not work.
Not operational (destination not active)	The destination interface is not in the Up state. See the corresponding *status* 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 the **show interfaces** command. Check the configuration to see wh be keeping the interface from coming up (for example, a sub-interfa to have an appropriate encapsulation configured).
Error: see detailed output for explanation	Traffic mirroring has encountered an error. Run the **show monitor** **status detail** 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 trace ?

 platform  Enable platform trace
 process   Filter debug by process(cisco-support)

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

 errors  Display error traces(cisco-support)
 events  Display event traces(cisco-support)

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

usrtdir  Specify directory to collect unsorted traces(cisco-support)
|        Output Modifiers
<cr>

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

usrtdir  Specify directory to collect unsorted traces(cisco-support)
|        Output Modifiers
<cr>

#



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

RP/0/RP0/CPU0:router# show tech-support span ?

 file       Specify a valid file name (e.g. disk0:tmp.log)
 list-CLIs  list the commands that would be run (don't execute)(cisco-support)
 location   Specify a location(cisco-support)
 rack       Specify a rack(cisco-support)
 time-out   per show command timeout configuration(cisco-support)
 <cr>
```