SPAN to file

This chapter explains a traffic monitoring feature that captures traffic to PCAP files for later analysis. It details capabilities such as partial packet capture, transmit (Tx) direction mirroring, continuous Always-On capture with periodic writes, and Unique Capture to reduce redundant data.

SPAN to file

SPAN to file is a traffic monitoring feature that

- captures traffic from a SPAN session and writes it directly to a file for later analysis
- extends the existing SPAN feature by mirroring network packets to a file instead of an interface
- helps in the analysis of the packets at a later stage, and
- saves the file in the PCAP file format, which is compatible with tools like tcpdump and Wireshark.

The **monitor-session** <*name*> [**ethernet**|**ipv4**|**ipv6**|**mpls-ipv4**|**mpls-ipv6**] command creates a monitor-session with the specified name and traffic class, serving as a chain point from the SPAN feature.

The **destination file** [**size** <*kbytes*>] [**buffer-type linear**] command adds a **file** destination option to the session configuration.

destination file has these configuration options:

- Buffer size: Sets the maximum file size for captured packets.
- Buffer types:
  - Circular: This is the default buffer type. Once the buffer is full, it overwrites from the beginning.
  - Linear: Once the buffer is full, no further packets are logged. Only the linear buffer type must be explicitly configured.

Altering any parameters, such as buffer size or type, recreates the session and clears any packet buffers.

All configuration options available for other SPAN types should also be supported by SPAN to file. These include options such as applying ACLs and capturing only the first X bytes of each packet.

Starting with Cisco IOS XR Release 7.5.3, truncation is supported per global session rather than per interface. These options are implemented by the router when punting the packet.

Table 1: Feature History Table

Feature name	Release information	Feature description
Partial packet capture ability for SPAN-to-file (Rx)	Release 25.2.1	Introduced in this release on: Centralized Systems (8400 [ASIC:K100]))(select variants only*) This feature allows to perform partial packet capture, also known as truncation, in the Rx direction. *This feature is now supported on Cisco 8404-SYS-D routers.
SPAN-to-file in Tx direction	Release 25.2.1	Introduced in this release on: Centralized Systems (8400 [ASIC:K100]))(select variants only*) This feature provides the ability to capture the packet in the Tx direction, store the capture on the file, and analyze the outgoing (Tx) packets. *This feature is now supported on Cisco 8404-SYS-D routers.
SPAN-to-File with unique capture	Release 25.2.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) This feature enhances the SPAN-to-File functionality by allowing you to capture only a single, unique packet for each punt reason or interface. This prevents interesting packets from being overshadowed by repeated packets in the analysed flow, ensuring that diverse and relevant packets are retained for analysis. The feature introduces these changes: CLI: The **unique-punt** and **unique-port** keywords are introduced in the **drops** command.

Feature name	Release information	Feature description
Always-On SPAN-to-File with periodic write	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]). The routers can now provide reliable, always-available packet capture for post-event analysis, eliminating the need for prior configuration or user interaction. The enhanced SPAN-to-File feature provides continuous packet capture and debugging capability with always-on functionality that starts automatically upon destination configuration. It prevents data loss during node reloads by periodically writing packet buffer contents to disk, without stopping the capture. A default SPAN-to-File session for forwarding and buffer drops is always active and can be disabled if not needed. The feature also supports packet truncation and sampling in software for software-mirrored packets, independent of NPU capabilities. The feature introduces these changes: CLI: • **monitor-session default-capture-disable** • **monitor-session local-capture-capacity** • The **always-on, periodic-write,**and **capacity** keywords are introduced in the **destination file** command. • The **write** keyword is introduced in the **monitor-session <name> packet-collection** action command. YANG data models: • New Xpaths for `Cisco-IOS-XR-um-monitor-session-cfg.yang` • New Xpaths for `Cisco-IOS-XR-Ethernet-SPAN-cfg.yang` • New Xpaths for `Cisco-IOS-XR-Ethernet-SPAN-act.yang` (see GitHub, YANG Data Models Navigator)
SPAN-to-file in Tx direction	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I

Feature name	Release information	Feature description
SPAN-to-file in Tx direction	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700)(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*). This feature now allows to capture packets in the Tx direction on the following hardware. *This feature is now supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 88-LC1-36EH
SPAN-to-file support in Tx and Rx direction	Release 7.5.3	With this feature, the ability to capture the packet in Tx direction along with the ability to store the capture on the file is supported. You can now capture the packet in the Tx direction and store the capture on the file. Earlier, you could only capture or mirror the traffic in the Rx direction. You now have the flexibility to choose Tx, Rx, or both directions. You can now capture and analyze the outgoing (Tx) packets.
Partial packet capture ability for SPAN-to-file (Rx)	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I

Feature name	Release information	Feature description
Partial packet capture ability for SPAN-to-file (Rx)	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700)(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*). This feature now allows you to perform partial packet capture in the Rx direction on the following hardware. *This feature is now supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 88-LC1-36EH
Partial packet capture ability for SPAN-to-file (Rx)	Release 7.5.3	With this feature, you can perform partial packet capture in the Rx direction. Earlier, the ability for entire packet capture was available in the Tx direction only, now you can choose entire or partial packet capture in the Rx direction also. Here, partial packet capture is also known as truncation.
SPAN-to-file PCAPng file format	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700)(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*). This PCAPng File Format feature that contains different blocks used to rebuild the captured packets into recognizable data is now supported on the following hardware. *This feature is now supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 88-LC1-36EH

Feature name	Release information	Feature description
SPAN-to-file PCAPng file format	Release 7.3.1	PCAPng is the next generation of packet capture format that contains a dump of data packets captured over a network and stored in a standard format. The PCAPng file contains different types of information blocks, such as the section header, interface description, enhanced packet, simple packet, name resolution, and interface statistics. These blocks can be used to rebuild the captured packets into recognizable data. The PCAPng file format: • Provides the capability to enhance and extend the existing capabilities of data storage over time • Allows you to merge or append data to an existing file. • Enables to read data independently from network, hardware, and operating system of the machine that made the capture.

This table details the support for packet capture for various releases:

Before Release 7.5.3	After Release 7.5.3
There was no functionality that you could enable to capture the payload of packets coming from your customers for security reasons.	The capture of all the outgoing packets from the router is supported.

Configuration guidelines for SPAN to file

These guidelines apply to SPAN to file:

- A maximum of 100 source ports are supported across the system. Individual platforms may support lower numbers.
- All the SPAN sessions are configured under the Ethernet class.
- At any given time, the system supports four SPAN to file sessions.
- When you attach multiple interfaces to a monitor session, the minimum buffer size is 1KB. The maximum buffer size is 1000KB, and default buffer size is 2KB.

Restrictions for SPAN to file

These restrictions apply to SPAN to file:

- Only incoming packet mirroring on the source interface is supported. Outgoing mirrored packets cannot be saved to a file. Starting with Cisco IOS XR Software Release 7.5.3, there are no restrictions.
- You can apply SPAN ACLs only in ingress direction. Hence, ACLs for SPAN to file can only be applied in ingress direction.

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

  ios_msg example:

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

- For outgoing (Tx) SPAN to file, security ACL is not supported.
- For outgoing (Tx) SPAN to file, only transit traffic is mirrored.
- Self-originating traffic cannot be mirrored.

Capabilities of SPAN to file

These are the supported capabilities of SPAN to file:

Traffic types and interface support

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

Direction and mirroring control

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

Truncation configuration

- Configure a different truncation size on each monitor session.
- Configure SPAN to file mirroring packet truncation size from 1 to 10,000 bytes. Values outside this range are rejected with an error.
- Change the truncation size when packet collection has stopped; removing or re-adding the monitor session is not required.
- Change the truncation size during packet collection without stopping the monitor session.
- Mirror the entire packet by default if no truncation size is configured. If the packet size is smaller than the truncation size, the entire packet is mirrored.

How SPAN to file works

Summary

SPAN to file works by capturing traffic from a SPAN session and saving the data to a file for later analysis.

Workflow

These stages describe how SPAN to file works when you configure a single file as a destination for a SPAN session:

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

These stages describe how SPAN to file works when you attach multiple interfaces to a monitor session:

- Multiple interfaces are attached to a session.
- The system sends the packets to the interfaces on the same node.
- The packets are saved to the same file. Bundle interfaces can be attached to a session with a file destination, which is similar to attaching individual interfaces.

Action commands for SPAN to file

Action commands for SPAN to file are capabilities that:

- allow you to start and stop network packet collection
- write the contents of a packet buffer to disk without stopping the packet capture for all the sessions with active packet collection
- run on sessions where the destination is a file, and
- autocomplete names of the globally configured SPAN to file sessions.

This table details the action commands for SPAN to file:

Table 2: Action commands for SPAN to file

Action	Command	Description
Start	`monitor-session <name> packet-collection start`	Use this command to start writing packets for the specified session to the configured buffer. This command has no effect for sessions configured as `always-on`.

Action	Command	Description
Stop	`monitor-session <name> packet-collection stop [ discard-data | write directory <dir> filename <filename> ]`	Use this command to stop writing packets to the configured buffer. If you specify the `discard-data` option, the system clears the buffer. If you specify the `write` option, the system writes the buffer to disk before clearing it. When writing the buffer to disk, save the file in .pcap format at the following location: `/<directory>/<node_id>/<filename>`. If you include a .pcap extension when specifying the filename, the system will remove it to prevent the extension from being added twice. This command returns an error for sessions configured as `always-on`.
Write	`monitor-session <name> packet-collection write [directory <dir>] [filename <filename>]`	Use this command to write the contents of a packet buffer to disk without stopping the packet capture. The write command is available only for sessions with active packet collection. You can start the packet collection explicitly with the action command or through always-on collection. You may specify the full directory path and file name where the buffer needs to be written. If you specify the directory, it must already exist. If the directory or file name are not specified, the following default values are used: Directory: /misc/scratch/SPAN/<node>/ Filename: <session_name>_<node>_<timestamp>.pcap

Configure SPAN to file

Use these steps to configure SPAN to file:

Procedure

Step 1 Create a monitor session.

Example:

```
interface GigabitEthernet 0/0/0/0
monitor-session mon1 [ethernet|ipv4|ipv6|mpls-ipv4|mpls-ipv6]
```

Step 2 Configure a **mon1** monitor session.

Example:

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

In this example, omitting the **buffer-type** option results in default circular buffer.

Step 3 Configure a **mon2** monitor session.

Example:

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

Step 4 Attach monitor session to a physical or bundle interface.

Example:

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

Step 5 Verify the packet collection status.

Example:

```
Router#show monitor-session status
Monitor-session mon1
Destination File - Packet collecting
=============================================
Source Interface Dir. Status
--------------------- ---- ------------------
Hu0/9/0/2 Rx Operational
Monitor-session mon2
Destination File - Packet collecting
=======================================
Source Interface Dir Status
--------------------- ---- --------------
BE2.1. Rx Operational
```

If packet collection is not active, this line is displayed:

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

Configure SPAN to file for truncation

Follow these steps to configure SPAN to file for truncation.

Procedure

Step 1 Create a SPAN to file monitor session for mirroring the packets with truncation enabled. Use the **mirror first** command in monitor session configuration.

Example:

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

Step 2 Attach the interfaces to the monitor session.

Example:

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

Step 3 Verify the configuration.

Example:

```
Router#show monitor-session status
                  Monitor-session mon1
                  Destination File - Packet collecting
                  ============================================
                  Source Interface      Dir   Status
                  --------------------- ----  ------------------
                  Hu0/9/0/2             Rx    Operational
                  Monitor-session mon2
                  Destination File - Packet collecting
                  =======================================
                  Source Interface      Dir   Status
                  --------------------- ----  --------------
                  BE1                  Tx    Operational
                  Monitor-session mon3
                  Destination File - Packet collecting
                  =======================================
                  Source Interface      Dir   Status
                  --------------------- ----  --------------
                  BE2.1                Both    Operational
```

Configure SPAN to file for direction

Follow these steps to configure SPAN to file for truncation.

Procedure

Step 1 Create a SPAN to file monitor session.

Example:

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

Step 2 Attach the interfaces to the monitor session.

Example:

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

Step 3 Verify the configuration.

Example:

```
Router#show monitor-session status
                  Monitor-session mon1
                  Destination File - Packet collecting
                  ============================================
                  Source Interface      Dir   Status
                  --------------------- ----  ------------------
                  Hu0/9/0/2             Rx    Operational
                  Monitor-session mon2
                  Destination File - Packet collecting
                  =======================================
                  Source Interface      Dir   Status
                  --------------------- ----  --------------
                  BE1                   Tx    Operational
                  Monitor-session mon3
                  Destination File - Packet collecting
                  =======================================
                  Source Interface      Dir   Status
                  --------------------- ----  --------------
                  BE2.1                 Both    Operational
```

Always-On SPAN to file with periodic write

The Always-On SPAN to file is a traffic mirroring feature that

- serves as a more reliable tool in investigating unexpected packet drops and traffic blackholing, and
- allows diagnosis of issues without reproducing faults, changing configurations, or needing prior user interaction before the event.

Always-On SPAN to file functionalities

These are the key functionalities of Always-On SPAN to file:

- Default SPAN enablement: Enables a default SPAN to-file session for packet forwarding and buffer drops automatically, provided the platform supports it. The session is always active and periodically writes to the disk without stopping the capture, up to the maximum configured storage capacity limit. This functionality ensures continuous packet capture and storage.

  You can disable this session if it's not needed using the **monitor-session default-capture-disable** command.

- SPAN truncation and sampling: Allows for packet truncation and sampling for software-mirrored packets like SPAN to file, even if hardware support isn't available. This functionality enhances flexibility by enabling these operations within the software.
- Always-on SPAN to file: Automatically starts packet capture when the file destination is configured, without requiring additional action commands. This functionality ensures immediate and continuous packet capture.

  Use the **destination file always-on** command in monitor-session configuration mode to enable always-on packet capture.
- SPAN to file continuous capture: Provides the ability to write packet data to a file without stopping the ongoing packet capture. This capability ensures uninterrupted packet monitoring and data collection.

  Use the **monitor-session <session name> packet-collection write** `[directory <dir>] [filename <file>]` command to write the current packet buffer to a file without stopping packet collection. If you don't use the optional keywords `directory` and `filename`, the system writes the buffer contents to a file named <session_name>_<node location>_<timestamp>.pcap in a default capture directory.
- SPAN periodic file writing: Allows you to set a period after which the buffered SPAN to file packet data is automatically written to a file. This automatic writing prevents data loss if there is a system reload and ensures persistent storage of captured packets. The feature includes configurable limits to manage file storage effectively, ensuring user-written files remain intact and session-specific data management doesn't impact other sessions.

  Use the **destination file** `[size <kbytes>] [always-on [periodic-write <secs> [capacity [<num> <kB|MB|GB>]]]]` command in monitor-session configuration mode to set the file writing interval.

  When the periodic-write option is used, the contents of the buffer are written to a file named <session_name>_<node location>_<timestamp>.pcap, in a default capture directory `/misc/scratch/SPAN/<node>/`.

Storage capacity management and file retention rules

There are two configurable capacity options to manage the periodically written files.

- A per-session limit: The maximum storage capacity for the set of files captured periodically for an individual monitor session. When this limit is exceeded, the system automatically deletes the oldest files to make room for new ones. Depending on the newer file size, it may delete multiple older files. In cases where only a single file is captured, the file is written completely, even if its size exceeds the per-session capacity limit. The file remains stored until another file is captured for that session.

  When a new file is significantly larger than the previous captures, the system may delete all existing files, leaving only the new file. This ensures that the new file is saved in its entirety without any truncation.

  Use the **destination file [capacity <num> <kB|MB|GB>]** command to configure the per-session capacity limit.
- A global limit: The total storage capacity for all files captured by SPAN on disk. If this limit is exceeded, further write operations don’t happen. Stopping the write operation without deleting files protects periodic and user-triggered writes in the default directory that remain unmoved or uncleared, regardless of session.

  Use the **monitor-session local-capture-capacity <num> <kB|MB|GB>** command to configure the global capacity limit.

Note This capacity limit configuration applies only to files written in the default directory. Any files moved out of the default directory don't count toward this limit.

If you don't configure these capacity parameters, the system uses default values specific to the platform variant to manage storage capacity.

Benefits of Always-On SPAN to file with periodic write

These are some of the benefits of the Always-On SPAN to file feature.

- **Reliable diagnostics**: Allows investigation of unexpected packet drops and traffic blackholing without the need to reproduce fault scenarios or change configurations.
- **Continuous packet capture**: Ensures uninterrupted packet monitoring and data collection by continuously capturing and storing packets without stopping.
- **Immediate activation**: Configuring the file destination automatically starts packet capture, ensuring immediate and continuous monitoring. Activating immediately reduces the risk of missing important data because of user mistakes or delays.
- **Data loss prevention**: Periodically writes packet data to a file, preventing data loss if there are system reloads and ensuring persistent storage.
- **Efficient storage management**: Provides configurable storage limits for individual sessions and overall capture, managing space effectively without deleting user-written files.
- **User control**: Allows you to disable default SPAN sessions and configure storage limits, giving the control over the packet capture and storage settings.

Guidelines for Always-On SPAN to file with periodic write

These guidelines apply to Always-On SPAN to file with periodic write:

- The platform enables the default SPAN to file session only if it supports it.
- If you do not specify directory and filename options for the **packet-collection write** action command, the system saves buffer contents as `<session_name>_<node location>_<timestamp>.pcap` in the default capture directory `/misc/scratch/SPAN/<node>/`.
- Ensure that the user-specified directory already exists before you use it in the **packet-collection write** action command.
- The system enforces a per-session storage limit; exceeding it results in the deletion of the oldest captures.
- The system enforces a global storage limit for all captured files; exceeding this prevents further write operations.
- The system enforces the global storage limit only on the default capture directory.
- If you do not configure storage capacity limit parameters, the system uses platform-specific default values.

- When you change a SPAN-to-File session from **always-on** to **on-demand,** you must explicitly stop packet collection or write the packet buffer to enable on-demand operation.

Configure Always-On SPAN to file with periodic write

This section includes configuration for always-on SPAN to file with periodic write and default enablement.

Procedure

Step 1 (Optional) Disable the default SPAN to file session by using the **monitor-session default-capture-disable** command. Default SPAN to file session for packet forwarding and buffer drops is enabled automatically.

Example:

```
Router#configure
Router(config)#monitor-session default-capture-disable
Router(config)#commit
```

Step 2 Enable Always-on SPAN to file by using the **destination file always-on** command in monitor-session configuration mode.

Example:

```
Router(config)#monitor-session test
Router(config-mon)#destination file always-on
Router(config-mon)#commit
```

Step 3 Write the current packet buffer to a file without stopping packet collection by using the **monitor-session <session name> packet-collection write** `[directory <dir>] [filename <file>]` action command.

Example:

```
Router#monitor-session test packet-collection write directory var/xr/scratch/SPAN/test file testfile
```

Step 4 Configure an interval for automatically writing buffered packet data to a file by using the **destination file** `[size <kbytes>] [always-on [periodic-write]]` command in monitor-session configuration mode.

Example:

```
Router(config)#monitor-session test
Router(config-mon)#destination file always-on periodic-write 300
Router(config-mon)#commit
```

Step 5 Set the storage capacity limit for all monitor-sessions by using the **monitor-session local-capture-capacity <num> <kB|MB|GB>** command in global configuration mode.

Example:

```
Router(config)#monitor-session local-capture-capacity 300 MB
Router(config)#monitor-session test
```

Step 6 Set the per-session limit by using the **destination file [capacity <num> <kB|MB|GB>]** command in monitor-session configuration mode, and save the changes.

Example:

```
Router(config-mon)#destination file always-on periodic-write 300 capacity 100 MB
Router(config-mon)#commit
```

Step 7 Verify the running configuration by using the **show running-config** command.

Example:

```
monitor-session test ethernet
 destination file always-on periodic-write 300 capacity 100 MB
!
monitor-session local-capture-capacity 300 MB
monitor-session default-capture-disable
```

Step 8 Verify the monitor-session details by using the **monitor-session status detail** command.

Example:

```
Router#show monitor-session status detail
Monitor-session test
  Destination File - Packet collecting (always-on)
  Periodic write interval: 300 seconds
  Maximum periodic capture capacity: 100MB
  Source Interfaces
  -----------------
......
```

Step 9 Verify global configuration items and information about platform capabilities by using the **show monitor-session status internal** command.

Example:

```
Router#show monitor-session status internal
Global Configuration:
    Router ID: Default
    Global local-capture-capacity: 300MB
    Default session disabled
Write command supported

Information from SPAN Manager and MA on all nodes:
Monitor-session test (ID 0x00000001) (Ethernet)
SPAN Mgr: Destination File - FileID:0
          Filename/directory name not set
          Last error: Success

Information from SPAN EA on all nodes:
Monitor-session 0x00000001 (Ethernet)
0/RP0/CPU0: Name 'test', destination file FileID:0
            Filename/directory name not set
Platform, 0/RP0/CPU0:

  Monitor Session ID: 1
  Truncation Size: 0
  Buffer type: Circular
  Buffer size: 2000
```

SPAN to file with unique capture

The SPAN to file with unique capture is a traffic mirroring feature that

- enables you to capture a single representative packet for each punt-reason or interface in a SPAN to file session,
- simplifies packet analysis

- reduces redundant data
- introduces a new buffer that ensures only one packet per port and punt reason is saved, and
- allows you to retain interesting packets while filtering out repetitive ones, making flow analysis more efficient.

SPAN to file with unique capture is implemented through the updated **monitor-session** command with options, such as **unique-punt** and **unique-port**.

Benefits of SPAN to file with unique capture

SPAN to file with unique capture offers several key benefits:

- Retains individual packets of interest, even in the presence of many repeated packets.
- Prevents important packets from being overwhelmed by repetitive traffic.
- Captures a variety of packet types for efficient and focused analysis.

How SPAN to file with unique capture works

Summary

Here's how SPAN to file with unique capture works:

- The SPAN to file with unique capture captures drop packets mirrored to the SPAN drop node and saves them to a PCAPng file.
- The feature ensures uniqueness by identifying packets based on either the punt reason or the interface (port).
- The process is managed by a database or table, ensuring only one packet is captured per punt reason or port in a unique buffer, separate from the main SPAN buffer.

Configure SPAN to file with unique capture

To configure SPAN-to-file with unique option, you must first create a monitor session using the **monitor-session** command. Choose the appropriate options to filter the drop traffic and specify whether to capture unique packets per punt reason, per port, or both.

Procedure

Step 1 Create a monitor session.

Example:

This example shows how to create a monitor-session.

```
Router(config)# monitor-session mon1 ethernet
Router(config-mon)#
```

Step 2 Configure monitor session to capture unique packets by using the **unique-punt** and **unique-port** options available within the **drops** command.

Example:

This example shows how to capture unique packets for each punt reason for packet-processing drops.

```
Router(config)# monitor-session mon1 ethernet
Router(config-mon)# destination interface tunnel-ip2
Router(config-mon)# drops packet-processing rx unique-punt
Router(config)# commit
```

Note
If neither packet-processing nor traffic-management is explicitly configured, the SPAN session will, by default, capture drops from both traffic-management and packet-processing.