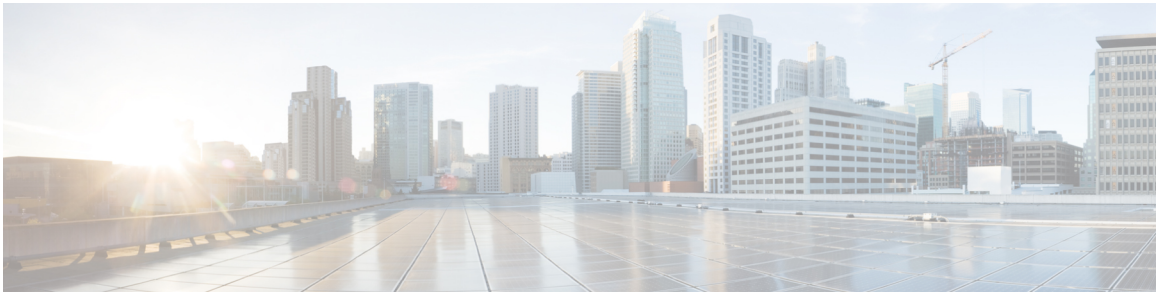
Troubleshooting Guide for Cisco Catalyst 8000V Edge Software

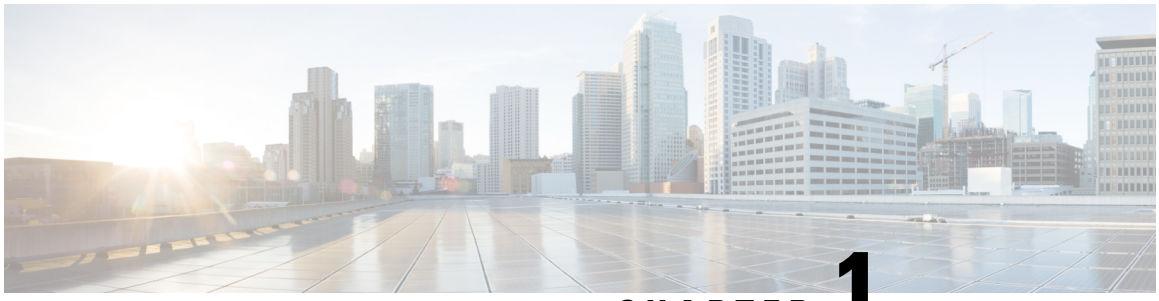
First Published: 2020-12-18

Last Modified: 2022-04-22

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS



CHAPTER 1

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

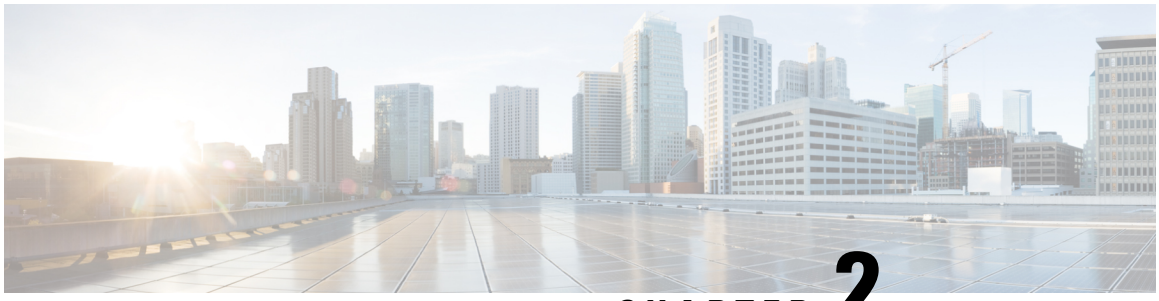
Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.



CHAPTER 2

Mapping the Cisco Catalyst 8000V Network Interfaces to VM Network Interfaces

Mapping the Router Network Interfaces to vNICs

Cisco Catalyst 8000V maps the GigabitEthernet network interfaces to the logical virtual network interface card (vNIC) name assigned by the VM. The VM in turn maps the logical vNIC name to a physical MAC address.

When you boot the Cisco Catalyst 8000V instance for the first time, the router interfaces are mapped to the logical vNIC interfaces that were added when the VM was created. The following image shows the relationship between the vNICs and the Cisco Catalyst 8000V router interfaces.

After you boot the Cisco Catalyst 8000V instance, you need to display the mapping between the logical interface on the router with the vNIC and the vNIC MAC address using the **show platform software vnic-if interface-mapping** command. The output for this command depends on your Cisco IOS XE release version.

Note GigabitEthernet0 interface is no longer supported.

```
Router# show platform software vnic-if interface-mapping
------------------------------------------------------------------------------
Interface Name        Short Name     vNIC Name                 Mac Addr
------------------------------------------------------------------------------
GigabitEthernet2       Gi2        eth2 (vmxnet3)           0050.5689.0034
GigabitEthernet1       Gi1        eth1 (vmxnet3)           0050.5689.000b
------------------------------------------------------------------------------
```

The vNIC name shown in the display is a logical interface that the Cisco Catalyst 8000V instance uses to map to the interface on the hypervisor. It does not always map to the corresponding NIC name added during the VM installation. For example, the logical “eth1” vNIC name in the display may not necessarily map to “NIC1” that was added in the VM installation process.

Caution It is important that you verify the interface mapping before you begin configuring the Gigabit Ethernet network interfaces onCisco Catalyst 8000V. This ensures that the network interface configuration applies to the correct physical MAC address interface on the VM host.

If you reboot the router and do not add or delete any vNICs, the interface mapping remains the same as before. If you reboot the router and delete vNICs, ensure that the configuration for the remaining interfaces remains intact. For more information, see *Adding and Deleting Network Interfaces on Cisco Catalyst 8000V*.

Adding and Deleting Network Interfaces on Cisco Catalyst 8000V

Cisco Catalyst 8000V maps the router GigabitEthernet interfaces to the logical vNIC name assigned by the VM which in turn is mapped to a MAC address on the VM host. You can add or delete vNICs on the VM to add or delete GigabitEthernet interfaces on Cisco Catalyst 8000V. You can add vNICs while the router is active.

To delete a vNIC from the VM, you must first power down the VM. If you delete any vNICs, you must reboot the router. For more information about adding and deleting vNICs, see the VMware Documentation .

Note Interface hot add/delete is not supported on Cisco Catalyst 8000V that operates in the Controller mode. If you need to perform interface hot add/delete, configure the reset operation in controller mode using the CLI: **`request platform software sdwan config reset`**.

Caution If you remove a vNIC without first updating the Cisco Catalyst 8000V network interface configuration, you risk a configuration mismatch when the router reboots. When you reboot the router and remove a vNIC, the remaining logical vNIC names could get reassigned to different MAC addresses. As a result, the GigabitEthernet network interfaces on theCisco Catalyst 8000V instances can be reassigned to different physical interfaces on the hypervisor.

Before you add or delete network interfaces, first verify the interface-to-vNIC mapping using the **show platform software vnic-if interface-mapping** command.

```
Router# show platform software vnic-if interface-mapping
------------------------------------------------------------------------
Interface Name             Driver Name              Mac Addr
------------------------------------------------------------------------
GigabitEthernet3            vmxnet3                 000c.2946.3f4d
GigabitEthernet2            vmxnet3                 0050.5689.0034
GigabitEthernet1            vmxnet3                 0050.5689.000b
------------------------------------------------------------------------
```

After adding or deleting network interfaces on the VM, verify the new interface-to-vNIC mapping before making configuration changes to the network interfaces. The following example shows the interface mapping after a new vNIC has been added. The new vNIC maps to the GigabitEthernet4 network interface on the Cisco Catalyst 8000V instance.

```
Router# show platform software vnic-if interface-mapping
```

```
-------------------------------------------------------------------------
Interface Name             Driver Name              Mac Addr
-------------------------------------------------------------------------
GigabitEthernet4            vmxnet3                0010.0d40.37ff
GigabitEthernet3            vmxnet3                000c.2946.3f4d
GigabitEthernet2            vmxnet3                0050.5689.0034
GigabitEthernet1            vmxnet3                0050.5689.000b
-------------------------------------------------------------------------
```

Removing a vNIC from a Running VM

To remove a vNIC from a running VM, use the `clear platform software` command (described below). Perform this command before removing a vNIC from the hypervisor configuration. This is part of a "two-step hot remove".

To see which hypervisors support a two-step hot remove, look for hypervisors with vNIC Two-Step Hot Remove Support = Yes

clear platform software vnic-if interface GigabitEthernet*interface-number*

interface-number - value from 0–32.

Example:

```
Router# clear platform software vnic-if interface GigabitEthernet4
```

Next, remove the vNIC from the hypervisor configuration.

Note You no longer need to execute the `clear platform software vnic-int interface` command before you remove the vNIC configuration from the hypervisor. This command will be deprecated in a future release.

Cisco Catalyst 8000V Network Interfaces and VM Cloning

When you first install a Cisco Catalyst 8000V instance, a database that maps the vNIC name to the MAC address is created. This database is used to maintain a persistent mapping between the router interfaces and the vNIC-to-MAC address mapping in case you add or delete vNICs. The interfaces are mapped to the stored Universal Unique Identification (UUID) maintained by VMware.

The mapping between the router network interfaces and the vNICs only applies to the current VM that the Cisco Catalyst 8000V is installed on. If the VM is cloned, the stored UUID will not match the current UUID and the interface mapping will not match the router configuration.

To prevent the interface mapping from becoming mis-matched, perform the following steps on the original VM before cloning:

Note Ensure that the original VM includes the number of configured vNICs required on the cloned VM before beginning the cloning process.

Step 1 Enter the **clear platform software vnic-if nvtable** command on the original VM.

This command clears the persistent interface database on the original VM and updates the interface mapping to the hypervisor.

Step 2 Reboot the Cisco Catalyst 8000V.

Step 3 On the cloned VM, verify the interface mapping using the **show platform software vnic-if interface-mapping** command.

Step 4 Configure the router interfaces on the cloned VM accordingly.

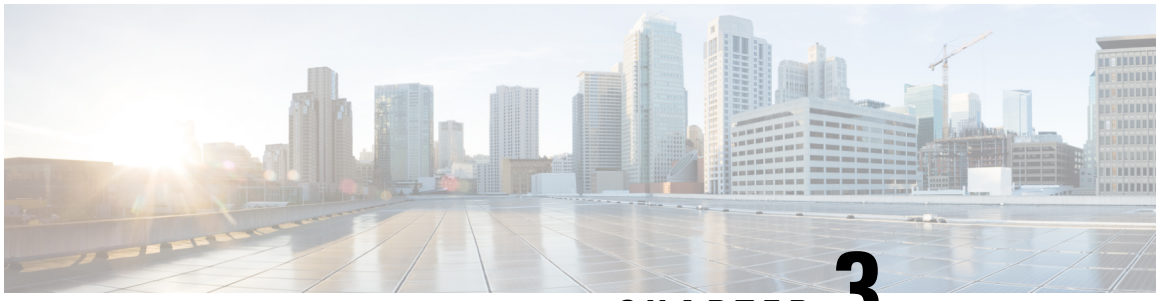
The router configuration on the cloned VM should match the configuration of the original VM.

Mapping the Cisco Catalyst 8000V Network Interfaces with vSwitch Interfaces

You can configure the network interfaces in ESXi in different ways to accommodate the Cisco Catalyst 8000V interfaces. You can configure the network interfaces so that each Cisco Catalyst 8000V router interface is mapped to one host Ethernet interface.

Alternatively, you can also configure the network interfaces so that multiple Cisco Catalyst 8000V interfaces share one host ESXi Ethernet interface.

The third possibility is mapping the Cisco Catalyst 8000V interfaces directly to a trunk interface on the vSwitch.



CHAPTER 3

Troubleshooting VM Issues

Troubleshooting Network Connectivity Issues

To troubleshoot network connectivity issues for Cisco Catalyst 8000V, do the following:

- Verify that there is an active and unexpired license installed on the VM.

Enter the **show license** command. The License State should be shown as “Active, In Use”.

- Verify that the vNIC for the VMs are connected to the correct physical NIC, or to the proper vSwitch.
- If you're using virtual LANS (VLANs), ensure the vSwitch is configured with the correct VLAN.
- If you're using static MAC addresses or VMs that are cloned, make sure there are no duplicate MAC addresses.

  Duplicate MAC addresses can cause the Cisco Catalyst 8000V feature license to become invalidated, which will disable the router interfaces.

Troubleshooting VM Performance Issues

Cisco Catalyst 8000V operates within a set of supported VM parameters and settings to provide certain levels of performance that have been tested by Cisco. Use the vSphere Client to view data to troubleshoot VM performance. If you are using vCenter, you can view historical data. If you are not using vCenter, you can view live data from the host.

This is a list of troubleshooting tips for performance issues:

Troubleshooting - MTU

Verify that the router has the correct setting for maximum MTU.

By default, the maximum MTU on the router is 1500. To support jumbo frames, edit the default VMware vSwitch settings. For more information, see the VMware vSwitch documentation.

Note ESXi 5.0 supports a maximum MTU of 9000, even if jumbo frames are enabled on the router.

Troubleshooting—Memory

Cisco Catalyst 8000V does not support memory sharing between VMs. On the ESXi host, check the memory counters to find out how much used memory and shared memory is on the VM. Verify that the balloon and swap used counters are zero.

If a specific VM does not have enough memory to support Cisco Catalyst 8000V, increase the memory size of the VM. Insufficient memory on the VM or the host can cause the Cisco Catalyst 8000V console to hang and be non-responsive.

Note With troubleshooting performance issues, note that other VMs on the same host as the Cisco Catalyst 8000V can impact the performance of the Cisco Catalyst 8000V VM. Verify that the other VMs on the host are not causing memory issues that are impacting the Cisco Catalyst 8000V VM.

Troubleshooting - Network Packets

Verify that no network packets are being dropped. On the ESXi host, check the network performance and view the counters to measure the number of receive packets and transmit packets dropped.

Troubleshooting - Throughput

Verify the current maximum throughput level with the **show platform hardware throughput level** command.

Troubleshooting - Instruction Extensions

Some x86 processors support instruction extensions for performing certain cryptographic transforms. Using these instructions is more efficient than not using them. Cisco Catalyst 8000V detects at run-time if the instruction extensions are available and will use them if they are available. To determine if the extensions are available, enter the **show platform software system all** command.

If the output shows that "Crypto Supported" is "No", then Cisco Catalyst 8000V may not exhibit the expected throughput. This is an issue with either the underlying physical hardware or the hypervisor. Check to see if the underlying physical hardware is capable of exposing the extensions and also check to see if the hypervisor can expose the extensions.

If the output shows that "Crypto Supported" is "Yes", then Cisco Catalyst 8000V provides the expected throughput because the physical hardware and the hypervisor can expose the extensions.

In the following example, "Crypto Supported" is "Yes". Therefore the cryptographic transforms can use instruction extensions, and perform efficiently.

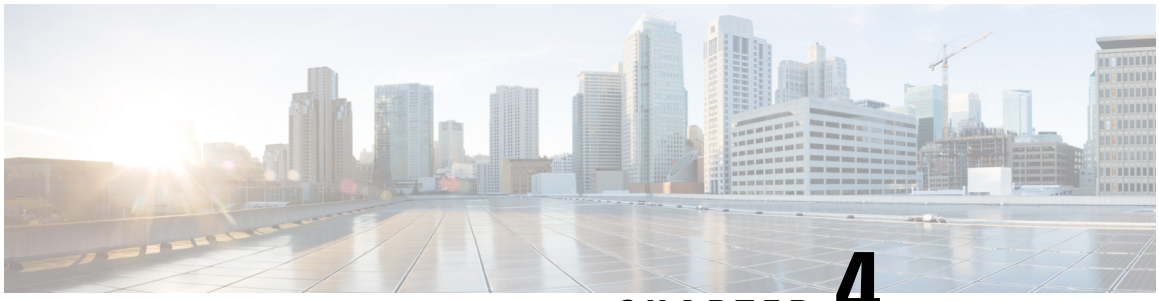
```
Router# show platform software system all
Processor Details
=================
```

```
Number of Processors : 4
Processor : 1 - 4
vendor_id : GenuineIntel
cpu MHz  : 3192.307
cache size : 20480 KB
Crypto Supported : Yes
```

IP address Inconsistency Issues on the vSphere Web Client

You might face inconsistencies in the IP addresses that is configured on the router and what is shown on the vSphere Web Client. At this moment there are no resolutions for this issue. See the following list to know why these inconsistencies might occur:

- ipv4 addresses for interfaces that are up or down are detected, while ipv6 addresses are only detected for interfaces that are up.
- After you perform an Interface Hot Delete, the vSphere Web Client continues to display the IP Address of the deleted interface.
- When you perform a reload on a Cisco Catalyst 8000V with addresses configured but not written to memory, the vSphere Web client continues to display the addresses even after the router comes up again. This occurs even though there are no addresses configured on the router. For example, configure Loopback, port-channel, port-group, and subinterfaces on a Cisco Catalyst 8000V router so that 63 addresses are displayed by the vSphere Web Client. Do not write the configuration to memory and reload the Cisco Catalyst 8000V. After the reload completes, all the 63 addresses are displayed on the Web Client. This occurs even though no addresses are configured on the Cisco Catalyst 8000V router. You can resolve this issue by configuring an address on the Cisco Catalyst 8000V router. When you do so, the web client then removes the 63 address and just displays the newly configured address.
- When you configure multiple ipv6 addresses on an interface, only the last address that you configured is detected. If you remove that address, none of the remaining configured ipv6 address on that interface are detected. This creates a state with multiple ipv6 addresses configured on an interface, but none displayed by the Web Client.
- When you delete interfaces, some of the addresses of the new interfaces are not displayed. This happens when the maximum number of IP Addresses are displayed and then you delete interfaces. For example, configure 32 Loopback interfaces with addresses and then delete each interface. Then, configure 32 GigabitEthernet sub interfaces with addresses. The addresses for the subinterfaces are not detected. This is because the router maintains entries for the deleted Loopback interfaces and is not able to add new interfaces.
- Addresses are detected for GigabitEthernet, Loopback, PortChannel, and VirtualPort-Group Interfaces as well as subinterfaces. However, Tunnel interface addresses are not detected.
- Secondary IP Addresses for IPv4 interfaces are not detected



CHAPTER 4

Packet Trace

First Published: August 03, 2016

The Packet-Trace feature provides a detailed understanding of how data packets are processed by the Cisco IOS XE platform, and thus helps customers to diagnose issues and troubleshoot them more efficiently. This module provides information about how to use the Packet-Trace feature.

Information About Packet Trace

The Packet-Trace feature provides three levels of inspection for packets: accounting, summary, and path data. Each level provides a detailed view of packet processing at the cost of some packet processing capability. However, Packet Trace limits inspection to packets that match the debug platform condition statements, and is a viable option even under heavy-traffic situations in customer environments.

The following table explains the three levels of inspection provided by packet trace.

Table 1: Packet-Trace Level

Packet-Trace Level	Description
Accounting	Packet-Trace accounting provides a count of packets that enter and leave the network processor. Packet-Trace accounting is a lightweight performance activity, and runs continuously until it is disabled.
Summary	At the summary level of packet trace, data is collected for a finite number of packets. Packet-Trace summary tracks the input and output interfaces, the final packet state, and punt, drop, or inject packets, if any. Collecting summary data adds to additional performance compared to normal packet processing, and can help to isolate a troublesome interface.

Packet-Trace Level	Description
Path data	The packet-trace path data level provides the greatest level of detail in packet trace. Data is collected for a finite number of packets. Packet-Trace path data captures data, including a conditional debugging ID that is useful to correlate with feature debugs, a timestamp, and also feature-specific path-trace data. Path data also has two optional capabilities: packet copy and Feature Invocation Array (FIA) trace. The packet-copy option enables you to copy input and output packets at various layers of the packet (layer 2, layer 3 or layer 4). The FIA- trace option tracks every feature entry invoked during packet processing and helps you to know what is happening during packet processing. **Note** Collecting path data consumes more packet-processing resources, and the optional capabilities incrementally affect packet performance. Therefore, path-data level should be used in limited capacity or in situations where packet performance change is acceptable.

Usage Guidelines for Configuring Packet Trace

Consider the following best practices while configuring the Packet-Trace feature:

- Use of ingress conditions when using the Packet-Trace feature is recommended for a more comprehensive view of packets.
- Packet-trace configuration requires data-plane memory. On systems where data-plane memory is constrained, carefully consider how you will select the packet-trace values. A close approximation of the amount of memory consumed by packet trace is provided by the following equation:

memory required = (statistics overhead) + number of packets * (summary size + data size + packet copy size).

When the Packet-Trace feature is enabled, a small, fixed amount of memory is allocated for statistics. Similarly, when per-packet data is captured, a small, fixed amount of memory is required for each packet for summary data. However, as shown by the equation, you can significantly influence the amount of memory consumed by the number of packets you select to trace, and whether you collect path data and copies of packets.

Configuring Packet Trace

Perform the following steps to configure the Packet-Trace feature.

Note The amount of memory consumed by the Packet-Trace feature is affected by the packet-trace configuration. You should carefully select the size of per-packet path data and copy buffers and the number of packets to be traced in order to avoid interrupting normal services. You can check the current data-plane DRAM memory consumption by using the **show platform hardware qfp active infrastructure exmem statistics** command.

SUMMARY STEPS

1. **enable**
2. **debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size***]**
3. **debug platform packet-trace {punt |inject|copy|drop|packet|statistics}**
4. **debug platform condition [ipv4 | ipv6] [interface** *interface***][access-list** *access-list -name* **|** *ipv4-address* **/** *subnet-mask* **|** *ipv6-address* **/** *subnet-mask***] [ingress | egress |both]**
5. **debug platform condition start**
6. **debug platform condition stop**
7. **show platform packet-trace {configuration | statistics | summary | packet {all |** *pkt-num***}}**
8. **clear platform condition all**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password if prompted.
Step 2	**debug platform packet-trace packet** *pkt-num* **[fia-trace | summary-only] [circular] [data-size** *data-size***]** **Example:** `Router# debug platform packet-trace packets 2048 summary-only`	Collects summary data for a specified number of packets. Captures feature path data by default, and optionally performs FIA trace. *pkt-num*—Specifies the maximum number of packets maintained at a given time. **fia-trace**—Provides detailed level of data capture, including summary data, feature-specific data. Also displays each feature entry visited during packet processing. **summary-only**—Enables the capture of summary data with minimal details. **circular**—Saves the data of the most recently traced packets. *data-size*—Specifies the size of data buffers for storing feature and FIA trace data for each packet in bytes. When very heavy packet processing is performed on packets, users can increase the size of the data buffers if necessary. The default value is 2048.
Step 3	**debug platform packet-trace {punt |inject|copy|drop|packet|statistics}** **Example:** `Router# debug platform packet-trace punt`	Enables tracing of punted packets from data to control plane.

	Command or Action	Purpose
Step 4	**debug platform condition [ipv4 | ipv6] [interface** *interface*]**[access-list** *access-list -name* | *ipv4-address* / *subnet-mask* | *ipv6-address* / *subnet-mask*] **[ingress | egress |both]** **Example:** `Router# debug platform condition interface g0/0/0 ingress`	Specifies the matching criteria for tracing packets. Provides the ability to filter by protocol, IP address and subnet mask, access control list (ACL), interface, and direction.
Step 5	**debug platform condition start** **Example:** `Router# debug platform condition start`	Enables the specified matching criteria and starts packet tracing.
Step 6	**debug platform condition stop** **Example:** `Router# debug platform condition start`	Deactivates the condition and stops packet tracing.
Step 7	**show platform packet-trace {configuration | statistics | summary | packet {all |** *pkt-num***}}** **Example:** `Router# show platform packet-trace 14`	Displays packet-trace data according to the specified option. See {start cross reference} Table 21-1 {end cross reference} for detailed information about the **show** command options.
Step 8	**clear platform condition all** **Example:** `Router(config)# clear platform condition all`	Removes the configurations provided by the **debug platform condition** and **debug platform packet-trace** commands.
Step 9	**exit** **Example:** `Router# exit`	Exits the privileged EXEC mode.

Displaying Packet-Trace Information

Use these **show** commands to display packet-trace information.

Table 2: show Commands

Command	Description
show platform packet-trace configuration	Displays packet trace configuration, including any defaults.
show platform packet-trace statistics	Displays accounting data for all the traced packets.

Command	Description
show platform packet-trace summary	Displays summary data for the number of packets specified.
show platform packet-trace {all | *pkt-num***} [decode]**	Displays the path data for all the packets or the packet specified. The **decode** option attempts to decode the binary packet into a more human- readable form.

Removing Packet-Trace Data

Use these commands to clear packet-trace data.

Table 3: clear Commands

Command	Description
clear platform packet-trace statistics	Clears the collected packet-trace data and statistics.
clear platform packet-trace configuration	Clears the packet-trace configuration and the statistics.

Configuration Examples for Packet Trace

This section provides the following configuration examples:

Example: Configuring Packet Trace

This example describes how to configure packet trace and display the results. In this example, incoming packets to Gigabit Ethernet interface 0/0/1 are traced, and FIA-trace data is captured for the first 128 packets. Also, the input packets are copied. The **show platform packet-trace packet 0** command displays the summary data and each feature entry visited during packet processing for packet 0.

```
Router>
enable
Router# debug platform packet-trace packet 128 fia-trace
Router# debug platform packet-trace punt
Router# debug platform condition interface g0/0/1 ingress
Router# debug platform condition start
Router#! ping to UUT
Router# debug platform condition stop
Router# show platform packet-trace packet 0
Packet: 0           CBUG ID: 9
Summary
  Input     : GigabitEthernet0/0/1
  Output    : GigabitEthernet0/0/0
  State     : FWD
  Timestamp
    Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
    Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
Path Trace
  Feature: IPV4
    Source      : 192.0.2.1
```

```
    Destination : 192.0.2.2
    Protocol    : 1 (ICMP)
  Feature: FIA_TRACE
    Entry     : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
    Timestamp : 3685243309297
  Feature: FIA_TRACE
    Entry     : 0x82011a00 - IPV4_INPUT_DST_LOOKUP_CONSUME
    Timestamp : 3685243311450
  Feature: FIA_TRACE
    Entry     : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
    Timestamp : 3685243312427
  Feature: FIA_TRACE
    Entry     : 0x82004b68 - IPV4_OUTPUT_LOOKUP_PROCESS
    Timestamp : 3685243313230
  Feature: FIA_TRACE
    Entry     : 0x8034f210 - IPV4_INPUT_IPOPTIONS_PROCESS
    Timestamp : 3685243315033
  Feature: FIA_TRACE
    Entry     : 0x82013200 - IPV4_OUTPUT_GOTO_OUTPUT_FEATURE
    Timestamp : 3685243315787
  Feature: FIA_TRACE
    Entry     : 0x80321450 - IPV4_VFR_REFRAG
    Timestamp : 3685243316980
  Feature: FIA_TRACE
    Entry     : 0x82014700 - IPV6_INPUT_L2_REWRITE
    Timestamp : 3685243317713
  Feature: FIA_TRACE
    Entry     : 0x82000080 - IPV4_OUTPUT_FRAG
    Timestamp : 3685243319223
  Feature: FIA_TRACE
    Entry     : 0x8200e500 - IPV4_OUTPUT_DROP_POLICY
    Timestamp : 3685243319950
  Feature: FIA_TRACE
    Entry     : 0x8059aff4 - PACTRAC_OUTPUT_STATS
    Timestamp : 3685243323603
  Feature: FIA_TRACE
    Entry     : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT
    Timestamp : 3685243326183

Router# clear platform condition all
Router# exit
```

Linux Forwarding Transport Service (LFTS) is a transport mechanism to forward packets punted from the CPP into applications other than IOSd. This example displays the LFTS-based intercepted packet destined for binos application.

```
Router# show platform packet-trace packet 10
Packet: 10     CBUG ID: 52
Summary
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  State  : PUNT 55 (For-us control)
  Timestamp
    Start : 597718358383 ns (06/06/2016 09:00:13.643341 UTC)
    Stop : 597718409650 ns (06/06/2016 09:00:13.643392 UTC)
Path Trace
  Feature: IPV4
    Input  : GigabitEthernet0/0/0
    Output : <unknown>
    Source : 10.64.68.2
    Destination : 10.0.0.102
    Protocol : 17 (UDP)
      SrcPort : 1985
      DstPort : 1985
```

```
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : <unknown>
  Entry  : 0x8a0177bc - DEBUG_COND_INPUT_PKT
  Lapsed time : 426 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : <unknown>
  Entry  : 0x8a017788 - IPV4_INPUT_DST_LOOKUP_CONSUME
  Lapsed time : 386 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : <unknown>
  Entry  : 0x8a01778c - IPV4_INPUT_FOR_US_MARTIAN
  Lapsed time : 13653 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017730 - IPV4_INPUT_LOOKUP_PROCESS_EXT
  Lapsed time : 2360 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017be0 - IPV4_INPUT_IPOPTIONS_PROCESS_EXT
  Lapsed time : 66 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017bfc - IPV4_INPUT_GOTO_OUTPUT_FEATURE_EXT
  Lapsed time : 680 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017d60 - IPV4_INTERNAL_ARL_SANITY_EXT
  Lapsed time : 320 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017a40 - IPV4_VFR_REFRAG_EXT
  Lapsed time : 106 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017d2c - IPV4_OUTPUT_DROP_POLICY_EXT
  Lapsed time : 1173 ns
Feature: FIA_TRACE
  Input  : GigabitEthernet0/0/0
  Output : internal0/0/rp:1
  Entry  : 0x8a017940 - INTERNAL_TRANSMIT_PKT_EXT
  Lapsed time : 20173 ns
LFTS Path Flow: Packet: 10    CBUG ID: 52
  Feature: LFTS
  Pkt Direction: IN
  Punt Cause  : 55
       subCause : 0
```

Example: Using Packet Trace

This example provides a scenario in which packet trace is used to troubleshoot packet drops for a NAT configuration on a Cisco device. This example shows how you can effectively utilize the level of detail provided by the Packet-Trace feature to gather information about an issue, isolate the issue, and then find a solution.

In this scenario, you can detect that there are issues, but are not sure where to start troubleshooting. You should, therefore, consider accessing the Packet-Trace summary for a number of incoming packets.

```
Router# debug platform condition ingress
Router# debug platform packet-trace packet 2048 summary-only
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Pkt   Input             Output            State  Reason
0     Gi0/0/0           Gi0/0/0           DROP   402 (NoStatsUpdate)
1     internal0/0/rp:0  internal0/0/rp:0  PUNT   21  (RP<->QFP keepalive)
2     internal0/0/recycle:0  Gi0/0/0      FWD
```

The output shows that packets are dropped due to NAT configuration on Gigabit Ethernet interface 0/0/0, which enables you to understand that an issue is occurring on a specific interface. Using this information, you can limit which packets to trace, reduce the number of packets for data capture, and increase the level of inspection.

```
Router# debug platform packet-trace packet 256
Router# debug platform packet-trace punt
Router# debug platform condition interface Gi0/0/0
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Router# show platform packet-trace 15
Packet: 15          CBUG ID: 238
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:1
  State     : PUNT 55  (For-us control)
  Timestamp
    Start   : 1166288346725 ns (06/06/2016 09:09:42.202734 UTC)
    Stop    : 1166288383210 ns (06/06/2016 09:09:42.202770 UTC)
Path Trace
  Feature: IPV4
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.64.68.3
    Destination : 10.0.0.102
    Protocol    : 17 (UDP)
      SrcPort   : 1985
      DstPort   : 1985
IOSd Path Flow: Packet: 15    CBUG ID: 238
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From CPP
  Feature: IP
    Pkt Direction: IN
    Source      : 10.64.68.122
    Destination : 10.64.68.255
  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.64.68.122
    Destination : 10.64.68.255
    Interface   : GigabitEthernet0/0/0
  Feature: UDP
    Pkt Direction: IN
    src         : 10.64.68.122(1053)
    dst         : 10.64.68.255(1947)
    length      : 48
```

```
Router#show platform packet-trace packet 10
Packet: 10          CBUG ID: 10
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:0
  State     : PUNT 55  (For-us control)
  Timestamp
    Start   : 274777907351 ns (01/10/2020 10:56:47.918494 UTC)
    Stop    : 274777922664 ns (01/10/2020 10:56:47.918509 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.78.106.2
    Destination : 10.0.0.102
    Protocol    : 17 (UDP)
      SrcPort   : 1985
      DstPort   : 1985

IOSd Path Flow: Packet: 10    CBUG ID: 10
  Feature: INFRA
    Pkt Direction: IN
Packet Rcvd From DATAPLANE
 Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.78.106.2
    Destination : 10.0.0.102
    Interface   : GigabitEthernet0/0/0

  Feature: UDP
    Pkt Direction: IN DROP
    Pkt : DROPPED
    UDP: Discarding silently
    src        : 881 10.78.106.2(1985)
    dst         : 10.0.0.102(1985)
    length      : 60

Router#show platform packet-trace packet  12
Packet: 12          CBUG ID: 767
Summary
  Input     : GigabitEthernet3
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 16120990774814 ns (01/20/2020 12:38:02.816435 UTC)
    Stop    : 16120990801840 ns (01/20/2020 12:38:02.816462 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet3
    Output      : <unknown>
    Source      : 10.1.1.1
    Destination : 10.1.1.2
    Protocol    : 6 (TCP)
      SrcPort   : 46593
      DstPort   : 23
IOSd Path Flow: Packet: 12    CBUG ID: 767
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
```

```
    Source      : 10.1.1.1
    Destination : 10.1.1.2
    Interface   : GigabitEthernet3

  Feature: IP
    Pkt Direction: IN
    FORWARDEDTo transport layer
    Source        : 10.1.1.1
    Destination   : 10.1.1.2
    Interface     : GigabitEthernet3

  Feature: TCP
    Pkt Direction: IN
    tcp0: I NoTCB 10.1.1.1:46593 10.1.1.2:23 seq 1925377975 OPTS 4 SYN  WIN 4128
```

```
Router# show platform packet-trace summary
Pkt   Input                     Output                    State  Reason
0     INJ.2                     Gi1                       FWD
1     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
2     INJ.2                     Gi1                       FWD
3     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
4     INJ.2                     Gi1                       FWD
5     INJ.2                     Gi1                       FWD
6     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
7     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
8     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
9     Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
10    INJ.2                     Gi1                       FWD
11    INJ.2                     Gi1                       FWD
12    INJ.2                     Gi1                       FWD
13    Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
14    Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
15    Gi1                       internal0/0/rp:0          PUNT   11  (For-us data)
16    INJ.2                     Gi1                       FWD
```

The following example displays the packet trace data statistics.

```
Router#show platform packet-trace statistics
Packets Summary
  Matched  3
  Traced   3
Packets Received
  Ingress  0
  Inject   0
Packets Processed
  Forward  0
  Punt     3
    Count       Code  Cause
    3           56    RP injected for-us control
  Drop     0
  Consume  0

          PKT_DIR_IN
             Dropped       Consumed       Forwarded
INFRA            0              0              0
TCP              0              0              0
UDP              0              0              0
IP               0              0              0
IPV6             0              0              0
ARP              0              0              0

         PKT_DIR_OUT
             Dropped       Consumed       Forwarded
INFRA            0              0              0
TCP              0              0              0
```

```
UDP              0             0             0
IP               0             0             0
IPV6             0             0             0
ARP              0             0             0
```

The following example displays packets that are injected and punted to the forwarding processor from the control plane.

```
Router#debug platform condition ipv4 10.118.74.53/32 both
 Router#Router#debug platform condition start
Router#debug platform packet-trace packet 200
Packet count rounded up from 200 to 256

Router#show platform packet-tracer packet 0
show plat pack pa 0
Packet: 0           CBUG ID: 674
Summary
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 17756544435656 ns (06/29/2020 18:19:17.326313 UTC)
    Stop    : 17756544469451 ns (06/29/2020 18:19:17.326346 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Protocol    : 17 (UDP)
      SrcPort   : 2640
      DstPort   : 500

IOSd Path Flow: Packet: 0    CBUG ID: 674
  Feature: INFRA
  Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
  Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Interface   : GigabitEthernet1

  Feature: IP
  Pkt Direction: IN
  FORWARDED To transport layer
    Source        : 10.118.74.53
    Destination   : 172.18.124.38
    Interface     : GigabitEthernet1

  Feature: UDP
  Pkt Direction: IN
  DROPPED
 UDP: Checksum error: dropping
 Source      : 10.118.74.53(2640)
 Destination : 172.18.124.38(500)

Router#show platform packet-tracer packet 2
Packet: 2           CBUG ID: 2

IOSd Path Flow:
  Feature: TCP
```

```
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN  WIN 4128

  Feature: TCP
  Pkt Direction: OUT
  FORWARDED
 TCP: Connection is in SYNRCVD state
 ACK         : 2346709419
 SEQ         : 3052140910
 Source      : 172.18.124.38(22)
 Destination : 172.18.124.55(52774)


  Feature: IP
  Pkt Direction: OUTRoute out the generated packet.srcaddr: 172.18.124.38, dstaddr:
172.18.124.55

  Feature: IP
  Pkt Direction: OUTInject and forward successful srcaddr: 172.18.124.38, dstaddr:
172.18.124.55

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN  WIN 4128
Summary
  Input     : INJ.2
  Output    : GigabitEthernet1
  State     : FWD
  Timestamp
    Start   : 490928006866 ns (06/29/2020 13:31:30.807879 UTC)
    Stop    : 490928038567 ns (06/29/2020 13:31:30.807911 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : internal0/0/rp:0
    Output      : <unknown>
    Source      : 172.18.124.38
    Destination : 172.18.124.55
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 52774
  Feature: IPSec
    Result    : IPSEC_RESULT_DENY
    Action    : SEND_CLEAR
    SA Handle : 0
    Peer Addr : 10.124.18.172
    Local Addr: 10.124.18.172


Router#
```

Additional References

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at this URL: {start hypertext}http://www.cisco.com/go/mibs{end hypertext}

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	{start hypertext}http://www.cisco.com/cisco/web/support/index.html{end hypertext}

Feature Information for Packet Trace

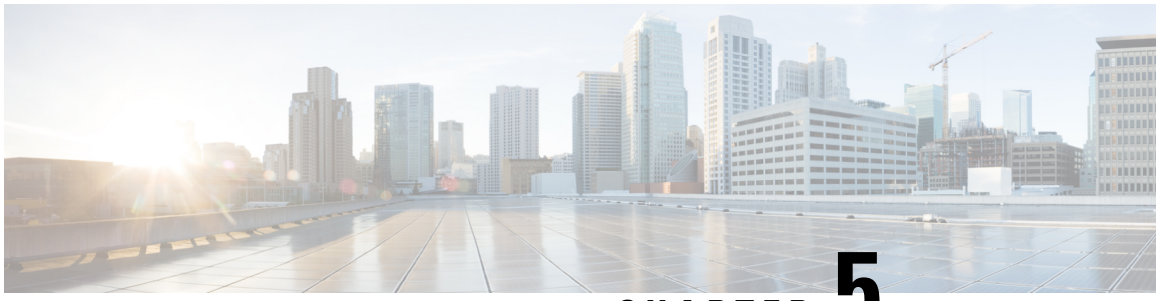
{start cross reference}Table 21-4{end cross reference} lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to{start hypertext} http://www.cisco.com/go/cfn{end hypertext}. An account on Cisco.com is not required.

Note {start cross reference}Table 21-4{end cross reference} lists only the software releases that support a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 4: Feature Information for Packet Trace

Feature Name	Releases	Feature Information
Packet Trace	Cisco IOS XE 3.10S	The Packet Trace feature provides information about how data packets are processed by the Cisco IOS XE software. In Cisco IOS XE Release 3.10S, this feature was introduced. The following commands were introduced or modified: • **debug platform packet-trace packet** *pkt-num* [**fia-trace** | **summary-only**] [**data-size** *data-size*] [**circular**] • **debug platform packet-trace copy packet** {**input** | **output** | **both**} [**size** *num-bytes*] [**L2** | **L3** | **L4**] • **show platform packet-trace** {**configuration** | **statistics** | **summary** | **packet** {**all** | *pkt-num*}}
	Cisco IOS XE 3.11S	In Cisco IOS XE Release 3.11S, this feature was enhanced to include the following features: • Matched versus traced statistics. • Trace stop timestamp in addition to trace start timestamp. The following commands were introduced or modified: • **debug platform packet-trace drop** [**code** *drop-num*] • **show platform packet-trace packet** {**all** | *pkt-num*} [*decode*]
	Cisco IOS XE Denali 16.3.1	In Cisco IOS XE Denali 16.3.1, this feature was enhanced to include Layer3 packet tracing along with IOSd. The following commands were introduced or modified: **debug platform packet-trace punt**.
	Cisco IOS XE Amsterdam 17.3.1	The output of the **show platform packet-trace** command now includes additional trace information for packets either originated from IOSd or destined to IOSd or other BinOS processes.



CHAPTER 5

Troubleshooting Packet Drops

Information About Troubleshooting Packet Drops

This chapter provides information on how to troubleshoot issues related to packet drops on Cisco Catalyst 8000V instances. From Cisco IOS XE 17.8.1, you can run the **show drops** command to troubleshoot the root cause of packet drops.

With the **show drop** command, you can identify:

- The reason for packet drop.
- The dropped interface with Rx or Tx direction.
- The root cause of the drop based on the feature or the protocol. You can filter the packet drop based on the interface, protocol, or feature.

- Viewing Packet Drop Information, on page 27
- Troubleshooting Packet Drops: Sample Output, on page 28

Viewing Packet Drop Information

Perform the following steps to view and filter the packet drop information for your instance based on the interface, protocol, or feature.

SUMMARY STEPS

1. **enable**
2. **show drops**
3. Show drops {bqs| crypto| firewall| interface| ip-all| nat| punt| qfp| qos}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**show drops** **Example:** `Router# show drops`	Displays the drop statistics for the instance.
Step 3	Show drops {bqs| crypto| firewall| interface| ip-all| nat| punt| qfp| qos} **Example:** `Router# show drops qfp`	Displays the drop statistics and the summary for the interface or the protocol that you choose.

Troubleshooting Packet Drops: Sample Output

The following is a sample output of the **show drops** command. This sample output displays the packet drops information related to the Quantum Flow Processor (QFP).

```
Router#show drops ?
  bqs        BQS related drops
  crypto     IPSEC related drops
  firewall   Firewall related drops
  interface  Interface drop statistics
  ip-all     IP related drops
  nat        NAT related drops
  punt       Punt path related drops
  qfp        QFP drop statistics
  qos        QoS related drops
  |          Output modifiers
  <cr>       <cr>

Router# show drops qfp
------------------ show platform hardware qfp active statistics drop detail ------------------

Last clearing of QFP drops statistics : Fri Feb 18 08:02:37 2022
 (6d 23h 54m 29s ago)
-------------------------------------------------------------------------------
   ID  Global Drop Stats                          Packets                 Octets
-------------------------------------------------------------------------------
  319  BFDoffload                                       9                   1350
   61  Icmp                                            84                   3780
   53  IpFragErr                                    32136                48718168
  244  IpLispHashLkupFailed                             3                    213
   56  IpsecInput                                      18                   4654
   23  TailDrop                                  26713208             10952799454
  216  UnconfiguredIpv6Fia                         241788                26596680

------------------ show platform hardware qfp active interface all statistics drop_summary
 ------------------

-----------------------------------------------------------------
Drop Stats Summary:
note: 1) these drop stats are only updated when PAL
         reads the interface stats.
      2) the interface stats include the subinterface

Interface                                     Rx Pkts             Tx Pkts
-------------------------------------------------------------------------
GigabitEthernet1                                60547                   0
```

```
GigabitEthernet2                          60782          27769658
GigabitEthernet3                          60581                 0
GigabitEthernet4                          60502           1323990
Tunnel14095001                                0           1990214
Tunnel14095002                                0           3883238
Tunnel14095003                                0           3879243
Tunnel14095004                                0           2018866
Tunnel14095005                                0           3875972
Tunnel14095006                                0           3991497
Tunnel14095007                                0           4107743
Tunnel14095008                                0           3990601
```