CHAPTER 4

Troubleshooting Packet Forwarding

This chapter explains how to troubleshoot router forwarding.

Cisco Express Forwarding (CEF) is the mechanism that enables packet forwarding. CEF information is examined when data forwarding is not occurring as expected. Troubleshooting CEF involves comparing the Routing Information Base (RIB) information to the software Forwarding Information Base (FIB), verifying that the hardware is programmed correctly, verifying that the adjacencies are built correctly, verifying the control plane is built correctly, and gathering any necessary trace information.

The only prerequisite for CEF is a valid route in the RIB.

This chapter includes the following sections:

Understanding IPv4 CEF

CEF is an advanced, Layer 3 IP switching technology that optimizes network performance. It also improves the scalability for networks with large and dynamic traffic patterns, such as the Internet and networks characterized by intensive Web-based applications.

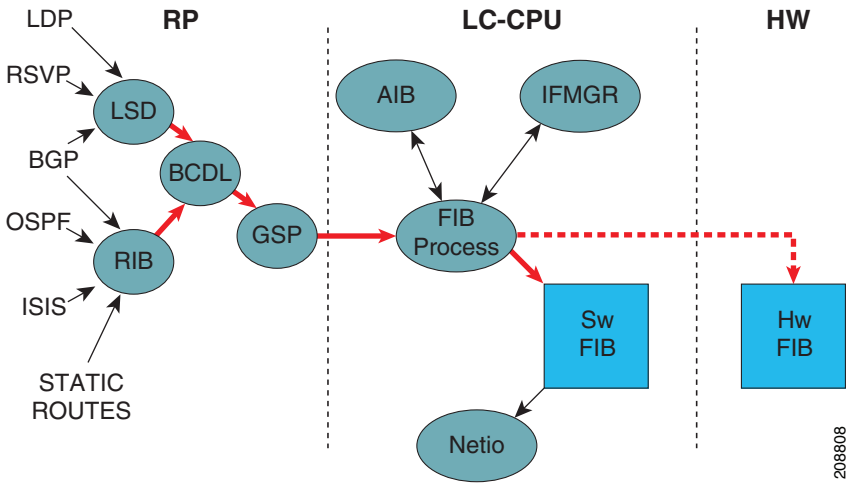
Information conventionally stored in a route cache is stored in several data structures for CEF switching. The data structures provide optimized lookup for efficient packet forwarding. The two main components of CEF operation are forwarding information base (FIB) and adjacency tables:

- CEF uses a FIB to make IP destination prefix-based switching decisions. FIB maintains a mirror image of the forwarding information contained in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next hop address information based on the information in the IP routing table. There is a one-to-one correlation between FIB entries and routing table entries, therefore FIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.
- Nodes in the network are said to be adjacent if they can reach each other with a single hop across a link layer. In addition to the FIB, CEF uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Figure 4-1 shows the components that contribute information to the CEF process, including autosynchronization of the RIB with the FIB.

Note In this document, the FIB is also referred to as the CEF table.

Figure 4-1 CEF Process



Troubleshooting IPv4 CEF

To troubleshoot IPv4 CEF information, perform the following procedure.

This procedure checks that neighbors are recognized, packets are flowing along the expected path, and packets are not being dropped between neighbor interfaces.

SUMMARY STEPS

1. **show route ipv4** *prefix*
2. **show cef ipv4** *prefix mask* **detail**
3. **show cef ipv4** *prefix mask* **detail location** *node-id* (on ingress line card)
4. **show cef ipv4** *prefix mask* **detail location** *node-id* (on egress line card)
5. **show cef ipv4** *prefix mask* **hardware ingress detail location** *node-id*
6. **show cef ipv4** *prefix mask* **hardware egress detail location** *node-id*
7. **show cef ipv4 interface** *type instance* **location** *node-id*
8. **show cef ipv4 summary location** *node-id*
9. **show cef ipv4 trace location** *node-id*
10. **show cef platform trace ipv4 all location** *node-id*
11. **show controllers pse qfp feature forward client ltrace unicast error location** *node-id*
12. Contact Cisco Technical Support if the problem is not resolved

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show route ipv4** *prefix* **Example:** `RP/0/RSP0/CPU0:router#  show route 192.168.2.0`	Displays the current routes in the Routing Information Base (RIB). • Check the prefix and mask, as well as the next hop and outgoing interface, to ensure that they are what is expected. • Note the timer value that shows how long the route has been in the routing table. If the timer value is low the route may be flapping. A lower timer value is present when a route is installed in the RIB for a short period of time. A low timer value may indicate flapping. For example, if a BGP route was being installed and removed from the RIB table every sixty seconds, then the route is flapping. Look for routes that have not been installed in the routing table for very long. The route will either be stable or flapping. If the route is flapping, contact contact Cisco Technical Support. For Cisco Technical Support contact information, see the "Obtaining Documentation and Submitting a Service Request" section on page xii in the Preface. • Check that route is learned via the routing protocol you are expecting, and that the metric is what you expect.
Step 2	**show cef ipv4** *prefix mask* **detail** **Example:** `RP/0/RSP0/CPU0:router#  show route ipv4 192.168.2.0 255.255.255.0 detail`	Displays the IPv4 Cisco Express Forwarding (CEF) table detailed entry information. • Compare the prefix, mask, next hop ip, and outgoing interface information with the information in the RIB. The information in the RIB is displayed using the **show route ipv4** *prefix mask* command as in Step 1. • Check that the adjacency is valid or the expected type of adjacency. For example, if it is a remote adjacency, then the adjacency information exists on another node. • Check that the expected hash (load balance) and egress interfaces are listed.

	Command or Action	Purpose
Step 3	**show cef ipv4** *prefix mask* **detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 detail location 0/14/cpu0`	Displays the IPv4 CEF table for the designated ingress node. • Compare the prefix, mask, next hop ip, and outgoing interface information with the information in the RIB. The information in the RIB is displayed using the **show route ipv4** *prefix* command as in Step 1. • Check that the adjacency is valid or the expected type of adjacency. For example, if it is a remote adjacency, then the adjacency information exists on another node. Check that the expected hash (load balance) and egress interfaces are listed.
Step 4	**show cef ipv4** *prefix mask* **detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 detail location 0/13/cpu0`	Displays the IPv4 CEF table for the designated egress node. • Compare the prefix, mask, next hop ip, and outgoing interface information with the information in the RIB. The information in the RIB is displayed using the **show route ipv4** *prefix mask* command. • Check that the adjacency is valid or the expected type of adjacency. For example, if it is a remote adjacency, then the adjacency information exists on another node. • Check that the expected hash (load balance) and egress interfaces are listed.
Step 5	**show cef ipv4** *prefix mask* **hardware ingress detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 hardware ingress detail location 0/14/cpu0`	Displays the IPv4 CEF table and corresponding forwarding chain for the designated ingress node. • Check that the prefix and mask are valid. • Check the nexthop IP address is as expected • Check that the entry type is set to forward. • Check the adjacency packet counter and byte counter.
Step 6	**show cef ipv4** *prefix mask* **hardware egress detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 192.168.2.0 255.255.255.0 hardware detail egress location 0/13/cpu0`	Displays the IPv4 CEF table and corresponding forwarding chain for the designated egress node. • Check that the prefix and mask are valid. • Check the nexthop IP address is as expected • Check that the entry type is set to forward. • Check the adjacency packet counter and byte counter.
Step 7	**show cef ipv4 interface** *type instance* **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 interface tengige 1/3/0/7 location 1/3/cpu0`	Displays IPv4 CEF-related information for an interface. Verify the interface handle 'interface is marked' is as expected. The command output also shows how many references there are to the interface in CEF table and the IPv4 MTU. Use this command for the ingress and egress interfaces.

	Command or Action	Purpose
Step 8	**show cef ipv4 summary location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 summary location 0/3/cpu0`	Displays a summary of the IPv4 CEF table. Check the VPN routing and forwarding (VRF) names associated with the node, the route update drops, and that there are the expected number of incomplete adjacencies. Note the number of routes CEF has entries for, the number of load sharing elements, and the number of references to this node. Use this command for the ingress and egress line cards and route processor (RP).
Step 9	**show cef ipv4 trace location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef ipv4 trace location 0/3/cpu0`	Displays IPv4 CEF trace table information. Check if there is any flap on the prefix. Use this command for the RP, and ingress and egress interfaces for the local line card.
Step 10	**show cef platform trace ipv4 all location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show cef platform trace ipv4 all location 0/3/cpu0`	Displays CEF IPv4 hardware status and configuration trace table information. Use this command for the ingress and egress interfaces for the local line card.
Step 11	**show controllers pse qfp feature forward client ltrace unicast error location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show contro pse qfp feature forward client ltrace unicast error location node-id`	(For SIP-700 line cards only) Displays trace files that contain information on any engine error (if any) that occurred in the unicast hardware structure programming.
Step 12	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the "Obtaining Documentation and Submitting a Service Request" section on page xii in the Preface.

Examples

The following examples show routes to two networks, one that is directly connected and one that is learned. In the first example, the route was installed about 19 days ago, which might be as expected. However, in the second example, the route was installed only 54 seconds ago, so it appears to be flapping:

```
RP/0/RSP0/CPU0:router# show route ipv4 10.114.4.11
Tue Jul 13 09:25:47.754 DST
Routing entry for 10.114.4.0/24
  Known via "connected", distance 0, metric 0 (connected)
  Installed Jul 12 14:18:06.668 for 19:07:41 <<< This route appears to be stable
  Routing Descriptor Blocks
    directly connected, via GigabitEthernet0/1/0/23
      Route metric is 0
  Redist Advertisers:
```

```
    ospf 100

RP/0/RSP0/CPU0:router# show route ipv4 10.119.4.19
Tue Jul 13 09:28:38.407 DST
Routing entry for 10.119.4.0/24
  Known via "ospf 100", distance 110, metric 2, type intra area
  Installed Jul 12 15:00:10.327 for 00:00:54 <<< This route appears to be flapping
  Routing Descriptor Blocks
    10.114.4.11, from 10.19.19.19, via GigabitEthernet0/1/0/23
      Route metric is 2
    10.114.8.11, from 10.19.19.19, via TenGigE0/4/0/0
      Route metric is 2
  No advertising protos.
```

The following examples show interface details.

```
RP/0/RSP0/CPU0:router# show cef ipv4 interface TenGigE 0/6/0/1 location 0/4/CPU0
Tue Jul 13 11:39:13.693 DST
UNKNOWN intf 0x00000001 is unknown if_handle 0x00000001 if_type 0x0
     idb info 0xa4d610d8 flags 0x301 ext 0xa5fe50cc
     Vrf Local Info (0x0)
  Interface last modified Jul 12, 2010 14:17:49, modify
  Interface is marked as point to point interface
  Reference count 1       Next-Hop Count 8
  Protocol Reference count 1
  Protocol ipv4 not configured or enabled on this card
  Primary IPV4 local address NOT PRESENT

RP/0/RSP0/CPU0:router# show cef ipv4 interface TenGigE 0/6/0/1 location 0/6/CPU0
Tue Jul 13 11:39:39.969 DST
TenGigE0/6/0/1 is down if_handle 0x100000c0 if_type 0x1e
     idb info 0xa4d61298 flags 0x1 ext 0x0
     Vrf Local Info (0x0)
  Interface last modified Jul 12, 2010 14:17:48, create
  Reference count 1       Next-Hop Count 0
  Protocol Reference count 0
  Protocol ipv4 not configured or enabled on this card
  Primary IPV4 local address NOT PRESENT
```

The following example shows the CEF summary. Use this display to check the VRF names, route update drops, and adjacencies:

```
RP/0/RSP0/CPU0:router# show cef ipv4 summary location 0/1/CPU0
Tue Jul 13 12:50:48.259 DST
Router ID is 10.144.144.144
IP CEF with switching (Table Version 552) for node0_1_CPU0
  Load balancing: L4
  Tableid 0xe0000000 (0xa4a6ddb0), Vrfid 0x60000000, Vrid 0x20000000, Flags 0x301
  Vrfname default, Refcount 251
  163 routes, 0 reresolve, 0 unresolved (0 old, 0 new), 13040 bytes
  60 load sharing elements, 129968 bytes, 342 references
  8 shared load sharing elements, 8564 bytes
  52 exclusive load sharing elements, 121404 bytes
  0 CEF route update drops, 0 CEF rcc update drops
  176 revisions of existing leaves
  Resolution Timer: 15s
  0 prefixes modified in place
  0 deleted stale prefixes
  99 prefixes with label imposition, 111 prefixes with label information
 23 next hops
  0 incomplete next hops
0 PD backwalks on LDIs with backup path
```

Troubleshooting Adjacency Information

To troubleshoot adjacency information on Cisco IOS XR software, perform the following procedure.

SUMMARY STEPS

1. **show arp location** *node-id*
2. **show arp traffic location** *node-id*
3. **show adjacency** *interface-type interface-instance* **remote detail location** *node-id*
4. **show adjacency** *interface-type interface-instance* **remote detail hardware location** *node-id*
5. **show adjacency ipv4 nexthop** *ipv4-address* **detail location** *node-id*
6. **show adjacency** *interface-type interface-instance* **detail location** *node-id*
7. **show adjacency ipv4 nexthop** *ipv4-address* **detail hardware location** *node-id*
8. **show adjacency** *interface-type interface-instance* **detail hardware location** *node-id*
9. **show adjacency trace location** *node-id*
10. **show adjacency trace client** *aib-client* **location** *node-id*
11. **show adjacency hardware trace location** *node-id*
12. **show cef adjacency tunnel-te** *tunnel-id* **hardware {egress | ingress} location** *node-id*
13. Contact Cisco Technical Support if the problem is not resolved

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show arp location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show arp location 0/12/cpu0`	Displays the Address Resolution Protocol (ARP) for an egress line card with a broadcast interface. Ensure that you can find the IP address and that correct MAC address of the neighbor is learned.
Step 2	**show arp traffic location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show arp traffic location 0/12/cpu0`	Displays ARP traffic statistics for an egress line card with a broadcast interface. Check for any errors or IP packet drops.
Step 3	**show adjacency** *interface-type interface-instance* **remote detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#  show adjacency pos 0/13/0/2 remote detail location 0/14/cpu0`	Displays detailed CEF adjacency table information for a remote ingress line card. Ensure that the output shows IPv4 adjacency information and that an adjacency exists.

	Command or Action	Purpose
Step 4	**show adjacency** *interface-type interface-instance* **remote detail hardware location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#  show adjacency pos 0/13/0/2 remote detail hardware location 0/14/cpu0`	Displays adjacency information for a remote ingress line card. • Check that the prefix and mask are valid. • Check that the table look-up (TLU) pointers match the TLU pointers in the **show cef ipv4** *prefix mask* **hardware ingress detail location** *node-id* command. For example:
Step 5	**show adjacency ipv4 nexthop** *ipv4-address* **detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#  show adjacency ipv4 nexthop 192.168.2.0 detail location 0/12/cpu0`	Displays adjacencies on an egress line card with a broadcast interface that are destined to the specified IPv4 next hop. When an egress interface is broadcast, use the **show adjacency ipv4 nexthop** command to display the adjacency information. Compare the mac layer rewrite information that shows the destination L2 address in the first part followed by the source L2 address, and the Ethernet value with the output from the **show arp location** *node-id* command.
Step 6	**show adjacency** *interface-type interface-instance* **detail location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#  show adjacency pos 0/13/0/2 detail location 0/13/cpu0`	Displays CEF adjacency table information for an egress line card with a point to point interface. There should be two IPv4 entries in the command output. Ensure both entries exist. • The SRC MAC only entry is used for multicast switching • The point to point entry is used for unicast switching. On broadcast interfaces you will have a SRC MAC only and one for each nexthop IP address. Please note the MTU is for the IPv4 minus the Layer 2 header. Use the **show im chains** command to display MTU details.
Step 7	**show adjacency ipv4 nexthop** *ipv4-address* **detail hardware location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#  show adjacency ipv4 nexthop 192.168.2.0 detail hardware location 0/12/cpu0`	Displays the hardware programming associated with the adjacency. Verify that the packets are being switched in the hardware.
Step 8	**show adjacency** *interface-type interface-instance* **detail hardware location** *node-id* **Example:** `RP/0/RSP0/CPU0:router#  show adjacency pos 0/13/0/2 detail hardware location 0/13/cpu0`	Displays the hardware programming information for a point-to-point interface such as the Packet-over-SONET/SDH (POS) interface. The rewrite information is slightly different because there is no MAC rewrite string as there is in Ethernet. Verify that the rewrite is appropriate for the encapsulation on the interface. Compare the CEF hardware output and verify that the pointer matches the egress adjacency.

	Command or Action	Purpose
Step 9	**show adjacency trace location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show adjacency trace location 0/1/cpu0`	Displays CEF adjacency trace table information. Use this command for the egress interfaces for the local line card.
Step 10	**show adjacency trace client** *aib-client* **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show adjacency trace client ipv4_fib_mgr location 0/13/cpu0`	Displays CEF adjacency trace table information for a specified adjacency information base (AIB) client. Use this command for the egress interfaces for the local line card.
Step 11	**show adjacency hardware trace location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show adjacency hardware trace location 0/13/cpu0`	Displays CEF adjacency hardware trace table information. Use this command for the egress interfaces for the local line card.
Step 12	**show cef adjacency tunnel-te** *tunnel-id* **hardware {egress | ingress} location** *node-id* **Example:** `show cef adjacency tunnel-te 1 hardware egress location 0/13/CPU`	Displays the IPv4 tunnel engineering (TE) tunnel adjacencies. Verify the tunnel adjacencies are as expected.
Step 13	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the "Obtaining Documentation and Submitting a Service Request" section on page xii in the Preface.

Examples

```
RP/0/RSP0/CPU0:router# show adjacency pos 0/2/0/1 remote detail hardware location 0/0/CPU0
Wed Nov  3 13:16:32.119 DST
Interface                  Address                  Version  Refcount Protocol
PO0/2/0/1                   (remote)                     15         1(    0) fint_n2n
                           040001c0
                           flags 1 0 2
                           0 packets, 0 bytes


RP/0/RSP0/CPU0:router# show cef 10.3.3.3 hardware ingress location 0/2/CPU0
Wed Nov  3 13:19:23.263 DST
10.3.3.3/32, version 0, internal 0x40040001 (ptr 0xa667ad70) [1], 0x0 (0xa5728bc4), 0x4500
(0xa754df28)
 Updated Oct 12 18:26:50.344
 remote adjacency to GigabitEthernet0/1/0/23
 Prefix Len 32, traffic index 0, precedence routine (0)
   via 10.114.4.11, GigabitEthernet0/1/0/23, 10 dependencies, weight 0, class 0 [flags
0x0]
    path-idx 0
    next hop 10.114.4.11
    remote adjacency
     local label 16018      labels imposed {16012}
   via 10.114.8.11, TenGigE0/4/0/0, 12 dependencies, weight 0, class 0 [flags 0x0]
```

```
    path-idx 1
    next hop 10.114.8.11
    remote adjacency
     local label 16018      labels imposed {16012}

TBM Node Data:
Node (0x00000002):0 0x8952700d 0x00000004 0x00000000 0xf7ff0000
Node (0x89527010):1 0x8944c2dd 0x88f92d50 0x08888888 0x88888888
Node (0x8944c2f0):2 0x88f9453d 0x00000000 0x10000000 0x00000000
Node (0x88f94530):3 0x88f9454d 0x00000000 0x80000000 0x00000000
Node (0x88f94540):4 0x88f94555 0x00000000 0x01000000 0x00000000
Node (0x88f94550):5 0x00002020 0x00000000 0x88fccc60 0x88f96320

Hardware Leaf Data (0x88f94550):0x00002020 0x00000000 0x88fccc60 0x88f96320

IP Leaf Data:
  as:0   prefix_len:32
  for_us:0x0     dft_route:0x0
  real_intf:0x1          free1: 0x0
  hw_use_only: 0x0
  lspa_ptr: 0x0          oce_chain_p: 0x88f96320
  extre_fib_data_ptr: 0x88fccc60

Hardware Extended Leaf Data:
  fib_leaf_extension_length: 0   interface_receive: 0x0
  traffic_index_valid: 0x0       qos_prec_valid: 0x0
  qos_group_valid: 0x0   valid_source: 0x0
  traffic_index: 0x0     nat_addr: 0x0
  reserved: 0x0          qos_precedence: 0x0
  qos_group: 0x0         peer_as_number: 0
  path_list_ptr: 0x0
  connected_intf_id: 0x0         ipsub_session_uidb: 0xffffffff
  Path_list:
    urpf loose flag: 0x0
    List of interfaces:

OCE Loadbalance Data for ptr 0x88f96320:
  num_entries:2          level:0x1
  pad_1:0x0      l3_lbe_ptr:0x8942d140

  LBE Array for 0x8942d140
    Entry 0: oce_chain_p 0x88f975b0
    Entry 0: bgp_ipv4_next_hop_addr: 0x0
    Entry 1: oce_chain_p 0x88f96e40
    Entry 1: bgp_ipv4_next_hop_addr: 0x0

OCE Label Object Data for ptr 0x88f975b0:
  flags: 0x0             number of labels: 1
  protocol: 0            number bk labels: 0
  out labels: 0x3e92
  next_hw_oce_ptr: 0x88f97850    counter_ptr: 0x893e9720
  Stats for ptr 0x893e9720:
    byte count: 0        packet count: 0

OCE RX Adj Data for 0x88f97850:
  base: 37(CPP HW RX ADJ MPLS)   adj_flags: 0x0
  pd_16: 0x1005          pd_32: 0x2f
  output_uidb: 0x1fea    counters_ptr: 0x893dc8a0
  byte count: 0          packet count: 0

OCE Label Object Data for ptr 0x88f96e40:
  flags: 0x0             number of labels: 1
  protocol: 0            number bk labels: 0
  out labels: 0x3e92
```

```
  next_hw_oce_ptr: 0x88f97840    counter_ptr: 0x893e9750
  Stats for ptr 0x893e9750:
    byte count: 0        packet count: 0

OCE RX Adj Data for 0x88f97840:
  base: 37(CPP HW RX ADJ MPLS)   adj_flags: 0x0
  pd_16: 0x6013          pd_32: 0x1
  output_uidb: 0x1fd0    counters_ptr: 0x893dc8b0
  byte count: 0          packet count: 0
```

The following example shows that the address information matches. The addresses are indicated in bold.

```
RP/0/RSP0/CPU0:router# show arp location 0/1/cpu0

Address         Age        Hardware Addr   State      Type  Interface
10.27.50.157   02:08:34   0016.c761.f509  Dynamic    ARPA  TenGigE0/1/0/2

RP/0/RSP0/CPU0:router# show adjacency ipv4 nexthop 212.27.50.157 detail loccation 0/1/cpu0

Interface                  Address                 Version  Refcount  Protocol
TenGigE0/1/0/2             10.27.50.157               41         2     ipv4
                          0016c761f5090015fa9959890800
                          mtu: 1500, flags 0 0 0
                          2894 packets, 156876 bytes
                          0xffffffff

RP/0/RSP0/CPU0:router# show adjacency gigabitEthernet 0/1/0/1 remote detail hardware
location all
Wed Nov  3 13:10:23.519 DST
------------------------------------------------------------------------------
0/1/CPU0
------------------------------------------------------------------------------
Interface                  Address                 Version  Refcount Protocol
Gi0/1/0/1                  (remote)                      6         1(    0) fint_n2n
                          020000c0
                          flags 1 0 2
                          0 packets, 0 bytes
------------------------------------------------------------------------------
0/RSP1/CPU0
------------------------------------------------------------------------------
Interface                  Address                 Version  Refcount Protocol
------------------------------------------------------------------------------
0/RSP0/CPU0
------------------------------------------------------------------------------
Interface                  Address                 Version  Refcount Protocol
Gi0/1/0/1                  (remote)                      7         1(    0) fint_n2n
                          020000c0
                          flags 1 0 2


RP/0/RSP0/CPU0:router# show cef adjacency tunnel-te 1 hardware egress location 0/3/CPU0
Wed Nov  3 13:37:17.935 DST
Interface not found (tunnel-te1)

Display protocol is ipv4
Interface    Address                                      Type    Refcount

BE16.162                                                  special 2
            Interface: BE16.162 Type:  glean
            Interface Type: 0x19, Base Flags: 0x4400 (0x9e4e9bb0)
            Nhinfo PT: 0x9e4e9bb0, Idb PT: 0x9e3591d8, If Handle: 0x80001a0
            Dependent adj type: remote (0x9f8af79c)
            Dependent adj intf: BE16.162
```

```
              Ancestor If Handle: 0x0


BE16.163                                                    special 2
              Interface: BE16.163 Type:  glean
              Interface Type: 0x19, Base Flags: 0x4400 (0x9e4e9d1c)
              Nhinfo PT: 0x9e4e9d1c, Idb PT: 0x9e359218, If Handle: 0x80001e0
              Dependent adj type: remote (0x9f8b033c)
              Dependent adj intf: BE16.163
              Ancestor If Handle: 0x0


tt44190       Prefix: 0.0.0.0/32                            local   3
              no next-hop adj
              Interface: NULLIFHNDL
 --More--
```

Troubleshooting Transient Traffic Drop

Perform this procedure to troubleshoot transient drops in packet forwarding. The approach to troubleshooting transient drops is as follows:

1. Determine the interface drops.
2. Determine the line card type. This is necessary because the next steps depend on whether you are troubleshooting an Ethernet or SIP-700 line card (LC).
3. (For Ethernet LC) Determine which NP contains the counters for the interface your are troubleshooting.
4. (For Ethernet LC) View the counters on the appropriate NP.
5. For SIP-700 LC, display the drop statistics on the LC.

SUMMARY STEPS

1. **show interface** *interface-type node-id*
2. **show platform**
3. **show controllers np ports all location** *node-id* (for Ethernet)
4. **show controllers np count** *np-id* **location** *node-id* (for Ethernet)
5. **show controllers pse qfp stat drop location** *node-id* (for SIP-700)

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interface** *interface-type node-id* **Example:** `show interface gigabitEthernet 0/0/0/0`	Displays the interface drops.
Step 2	**show platform**	Determines the line card type. This is necessary because the next steps depend on whether you are troubleshooting an Ethernet or SIP-700 line card (LC).

	Command or Action	Purpose
Step 3	**show controllers np ports all location** *node-id* **Example:**	(For Ethernet LC) Displays the port mapping between the interface and the NP. View the output and determine which NP contains the counters for the interface your are troubleshooting.
Step 4	**show controllers np count {***np-id* **| all} ocation** *node-id* **show controllers np count {***np-id* **all} location** *node-id* **| i DROP** **Example:** **show controllers np count all location 0/0/CPU0** **show controllers np count all location 00/0/CPU0 | i DROP**	(For Ethernet LC) View the counters on the appropriate NP. The first command displays all counters, whether related to drops or not. The second command limits the display to only those counters that include the string DROP. For additional information on interpreting NP counters, see the "Displaying Traffic Status in Line Cards and RSP Cards" section on page 7-147.
Step 5	**show controllers pse qfp stat drop location** *node-id* **Example:** **show controllers pse qfp stat drop location** *0/6/CPU0*	(For SIP-700 LC) Display the drop statistics on the LC.

Example

```
RP/0/RSP0/CPU0:router# show interface gigabitEthernet 0/0/0/0
Tue Oct 26 21:04:12.805 UTC
GigabitEthernet0/0/0/0 is up, line protocol is up
  Interface state transitions: 5
  Hardware is GigabitEthernet, address is 001b.53ff.a018 (bia 001b.53ff.a018)
  Internet address is 45.1.1.1/24
  MTU 2014 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, SXFD, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters 1w4d
  5 minute input rate 4000 bits/sec, 0 packets/sec
  5 minute output rate 11000 bits/sec, 0 packets/sec
     1590651 packets input, 551036131 bytes, 0 total input drops <<< drops by framer or HW
     97206 drops for unrecognized upper-level protocol <<< drops
     Received 0 broadcast packets, 332301 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort <<< drops
     1536152 packets output, 1427163508 bytes, 0 total output drops <<< sum of all output
drops, including drops from buffer, qos, or HW.
     Output 0 broadcast packets, 339069 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions


RP/0/RSP0/CPU0:router# show platform
```

```
Tue Oct 26 20:58:49.575 UTC
Node            Type                        State            Config State
--------------------------------------------------------------------------------
0/RSP0/CPU0     A9K-RSP-4G(Active)          IOS XR RUN       PWR,NSHUT,MON
0/0/CPU0        A9K-40GE-L                  IOS XR RUN       PWR,NSHUT,NMON <<< Ethernet
0/3/CPU0        A9K-8T/4-E                  IOS XR RUN       PWR,NSHUT,MON <<< Ethernet
0/4/CPU0        A9K-8T-E                    IOS XR RUN       PWR,NSHUT,MON <<< Ethernet
0/6/CPU0        A9K-SIP-700                 IOS XR RUN       PWR,NSHUT,NMON <<< SIP-700
0/6/0           SPA-2XCHOC12/DS0            OK               PWR,NSHUT,MON <<< SPA
0/6/1           SPA-5X1GE-V2                OK               PWR,NSHUT,MON <<< SPA


RP/0/RSP0/CPU0:router# show controllers np ports all loc 0/0/CPU0
Tue Oct 26 20:57:11.468 UTC

               Node: 0/0/CPU0:
----------------------------------------------------------------

NP Bridge Fia                      Ports
-- ------ --- -----------------------------------------------
0  0      0   GigabitEthernet0/0/0/30 - GigabitEthernet0/0/0/39
1  0      0   GigabitEthernet0/0/0/20 - GigabitEthernet0/0/0/29
2  1      0   GigabitEthernet0/0/0/10 - GigabitEthernet0/0/0/19
3  1      0   GigabitEthernet0/0/0/0 - GigabitEthernet0/0/0/9


RP/0/RSP0/CPU0:router# show controllers np counters all location 0/0/CPU0
Tue Oct 26 20:54:53.095 UTC

               Node: 0/0/CPU0:
----------------------------------------------------------------

Show global stats counters for NP0, revision v3

Read 23 non-zero NP counters:
Offset  Counter                                        FrameValue   Rate (pps)
--------------------------------------------------------------------------------
  22  PARSE_ENET_RECEIVE_CNT                       74772482296       60925
  23  PARSE_FABRIC_RECEIVE_CNT                        80571           0
  26  MODIFY_FABRIC_TRANSMIT_CNT                   36431746029       29685
  28  PARSE_INGRESS_DROP_CNT                          18816500           0

RP/0/RSP0/CPU0:router# show controllers np count all location 0/0/CPU0 | i DROP
Tue Oct 26 20:56:10.714 UTC
  28  PARSE_INGRESS_DROP_CNT                       38183944221           0
  30  RESOLVE_INGRESS_DROP_CNT                       157639443           0
  31  RESOLVE_EGRESS_DROP_CNT                           2559           0
 291  DROP_IPV4_NOT_ENABLED                        38174791832           0
 438  RESOLVE_MAC_NOTIFY_CTRL_DROP_CNT                  2559           0
  28  PARSE_INGRESS_DROP_CNT                          18816500           0
```

Note For a description of how to interpret NP counter information, see the "Displaying Traffic Status in Line Cards and RSP Cards" section on page 7-147.

```
RP/0/RSP0/CPU0:router# show controllers pse qfp stat drop location 0/6/CPU0
Tue Oct 26 20:57:49.864 UTC

Global Drop Statistics for QFP 0
----------------------------------------------------------------
Global Drop Stats                          Packets         Octets
----------------------------------------------------------------
```

```
AttnInvalidSpid                          0                0
BadAdj                                   0                0
BadBhdr                                  0                0
```

Troubleshooting Packet Drop in the Fabric

To check whether packets are being dropped in the fabric, use the following commands.

- **show controllers fabric fia bridge stats location** *node-id*
- **show controllers fabric fia drops ingress location** *node-id*
- **show controllers fabric fia drops egress location** *node-id*
- **show controllers fabric fia stats location** *node-id*

For detailed fabric troubleshooting procedures, see Chapter 7, "Troubleshooting Router Switch Fabric and Data Path."

Troubleshooting Control Plane Information

To troubleshoot control plane information on Cisco IOS XR software, perform the following procedure.

SUMMARY STEPS

1. **show netio idb** *interface-type interface-instance* **location** *node-id*
2. **show uidb index**
3. **show uidb data location** *node-id interface-type interface-instance* {ingress | egress}
4. **show imds interface brief**
5. Contact Cisco Technical Support if the problem is not resolved

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show netio idb** *interface-type interface-instance* **location** *node-id* **Example:** `RP/0/RSP0/CPU0:router# show netio idb tengige0/0/0/0 location 0/0/cpu0`	Displays control plane information for the software switching path. The output provides useful statistics for determining software forwarding issues. • Verify the encap and decap paths • Ensure that all the appropriate steps in the chain are shown for all the features that may be enabled on the interface. **Note** Fixup is a direct pointer to a routine in the output path after a CEF rewrite. this is an optimized path if a CEF rewrite exists and is usable. • Verify that the ifhandle and global uidb value is correct. Use this command for the ingress and egress interfaces for the local line card.
Step 2	**show uidb index** **Example:** `RP/0/RSP0/CPU0:router# show uidb index`	Displays the micro-interface descriptor block (IDB) index assigned by the software. Check that the interface and the universal interface descriptor block (UIDB) value are what is expected. Compare the IDB index to the uidb index value in the **show adjacency ipv4** *interface-type interface-instance* **detail hardware location** *node-id* command output.
Step 3	**show uidb data location** *node-id interface-type interface-instance* **{ingress | egress}** **Example:** `show uidb data location 0/6/CPU0 gigabitEthernet 0/0/0/2 ingress`	Displays, from a software perspective, features that are enabled on a selected interface. • Check the UIDB value. • Check what flags are enabled for the UIDB. • Check the ifhandle in the UIDB to make sure it is correct. Compare the output to the configuration of the interface and expected features. Use this command for the ingress and egress interfaces for the local line card.
Step 4	**show imds interface brief** **Example:** `RP/0/RSP0/CPU0:router# show imds interface brief`	Displays interface manager distribution server (IMDS) interface information. **Note** This is just a partial output not full output. Check the state, MTU, encapsulation being used, and the ifhandle for each interface.
Step 5	Contact Cisco Technical Support.	If the problem is not resolved, contact Cisco Technical Support. For Cisco Technical Support contact information, see the "Obtaining Documentation and Submitting a Service Request" section on page xii in the Preface.

Examples

The following example displays the control plane information for the software switching path. Check for any errors or drops.

```
RP/0/RSP0/CPU0:router# show netio idb tenGigE 0/1/1/0 location 0/1/cpu0

TenGigE0/1/1/0 (handle: 0x01180020, nodeid:0x11) netio idb:
---------------------------------
name:                     TenGigE0_1_1_0
interface handle:         0x01180020
interface global index:   2
physical media type:      30
dchain ptr:               <0x482ae8e0>
echain ptr:               <0x482d791c>
fchain ptr:               <0x482d79b8>
driver cookie:            <0x4824ad58>
driver func:              <0x4824ad44>
number of subinterfaces: 4096
subblock array size:      3
DSNCNF:                   0x00000000
interface stats info:
   IN  unknown proto pkts:  0
   IN  unknown proto bytes: 0
   IN  multicast pkts:      0
   OUT multicast pkts:      0
   IN  broadcast pkts:      0
   OUT broadcast pkts:      0
   IN  drop pkts:           0
   OUT drop pkts:           0
   IN  errors pkts:         0
   OUT errors pkts:         0

Chains
--------------------
Base decap chain:
     ether                <30>  <0xfd7aef88, 0x48302824>  <       0,        0>

Protocol chains:
---------------
<Protocol number> (name)  Stats
  Type  Chain_node       <caps num>  <function, context> <drop pkts, drop bytes>
<7> (arp)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     l2_adj_rewrite       <86>  <0xfcec7a88, 0x4834efec>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
    Decap:
     queue_fifo           <56>  <0xfcedda4c, 0x482dbee4>  <       0,        0>
     arp                  <24>  <0xfd1082cc, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite       <86>  <0xfcec745c, 0x00000000>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
<12> (ipv4)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     ipv4                 <26>  <0xfd10f41c, 0x482d7724>  <       0,        0>
     ether                <30>  <0xfd7aeb44, 0x48302824>  <       0,        0>
     l2_adj_rewrite       <86>  <0xfcec7a88, 0x4834f104>  <       0,        0>
     queue_fifo           <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull           <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
    Decap:
     queue_fifo           <56>  <0xfcedda4c, 0x482dbee4>  <       0,        0>
```

```
     ipv4                <26>  <0xfd10f474, 0x00000000>  <       0,        0>
    Fixup:
     l2_adj_rewrite      <86>  <0xfcec745c, 0x00000000>  <       0,        0>
     queue_fifo          <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull          <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
<22> (ether_sock)   Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     ether_sock          <98>  <0xfd7b1630, 0x48302824>  <       0,        0>
     l2_adj_rewrite      <86>  <0xfcec7a88, 0x48304c1c>  <       0,        0>
     queue_fifo          <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull          <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>
    Decap:
     queue_fifo          <56>  <0xfcedda4c, 0x482dbee4>  <       0,        0>
     ether_sock          <98>  <0xfd7b1874, 0x48302824>  <       0,        0>
    Fixup:
     l2_adj_rewrite      <86>  <0xfcec745c, 0x00000000>  <       0,        0>
     queue_fifo          <56>  <0xfcedda68, 0x482dbee4>  <       0,        0>
     txm_nopull          <60>  <0xfcea2a5c, 0x482dc11c>  <       0,        0>


Protocol SAFI counts:
--------------------

       Protocol        SAFI       Pkts In    Bytes In    Pkts Out   Bytes Out
---------------  ----------    ----------  ----------  ----------  ----------
           ipv4     Unicast             0           0           0           0
           ipv4   Multicast             0           0           0           0
           ipv4   Broadcast             0           0           0           0
           ipv6     Unicast             0           0           0           0
           ipv6   Multicast             0           0           0           0
```

The following example shows that the micro-idb index value is 12.

```
RP/0/RSP0/CPU0:router# show uidb index tengige1/3/0/6.30 location 1/3/cpu0

--------------------------------------------------------------------------------
  Location Interface-name      Interface-Type   Ingress-index  Egress-index
--------------------------------------------------------------------------------
  1/3/CPU0 TenGigE1_3_0_6.30   Sub-interface       20               12
```

Comparing the IDB index value of 12 in the **show uidb index** command to the uidb index value in the following command output shows that the values are the same.

```
RP/0/RSP0/CPU0:router# show adjacency ipv4 tengige1/3/0/6.30 detail hardware location
1/3/cpu0

Interface                  Address                Version  Refcount  Protocol
TenGigE1/3/0/6.30           (src mac only)             90         1     ipv4
                           00000000000001243602d8b8100001e0800
                           mtu: 1500, flags 1 0 1
                           453 packets, 42582 bytes
                           453 hw-only-packets, 42582 hw-only-bytes
 ether egress adjacency
  TLU1            : 0x4407
  [HW: 0x00401862 0xc4170800 0x8100001e 0x01060700]
    num. entries  : 1
    uidb index    : 12
    counter msb   : 0x2
    counter lsb   : 0xc417
    vlan e or len : 0x800
    ether len     : 0x8100 (33024)
    vlan info     : 30
    next ptr      : 0x10607
```

The following example displays, from a software perspective, features that are enabled on a selected interface. Compare the output to the configuration of the interface and expected features. Verify that the configured features are correctly enabled.

```
RP/0/RSP0/CPU0:router# show uidb data location 0/6/cpu0

------------------------------------------------------------------------------
  Location = 0/6/CPU0
  Index = 0
  Pse direction = INGRESS

  Global general 16 bytes:
  ------------------------
  ROUTER_ID: 45.104.151.108
  MINIMUM MASK DESTINATION: 0 / 0
  MINIMUM MASK SOURCE: 0 / 0
  BYTES OF SNIFF PACKET: 0
  SUPPRESS PUNT ACL: 0
  MPLS PROPAGATE TTL FLAG: 1
  PARITY: 0
  FABRIC QOS ENABLE FLAG: 0
------------------------------------------------------------------------------
  Location = 0/6/CPU0
  Index = 0
  Pse direction = EGRESS

  Global general 16 bytes:
  ------------------------
  ROUTER_ID: 45.104.151.108
  MINIMUM MASK DESTINATION: 0 / 0
  MINIMUM MASK SOURCE: 0 / 0
  BYTES OF SNIFF PACKET: 0
  SUPPRESS PUNT ACL: 0
  MPLS PROPAGATE TTL FLAG: 1
  PARITY: 0
  IPV4 PREFIX ACCNTG: 0
------------------------------------------------------------------------------
  Location = 0/6/CPU0
  Ifname/Ifhandle = GigabitEthernet0_6_5_0
  Index = 1
  Pse direction = INGRESS

  General 16 bytes:
  -----------------
  IFHANDLE: 0x168002
  STATUS: 0
  IPV4 ENABLE: 0
  IPV6 ENABLE: 0
  MPLS ENABLE: 0
  STATS POINTER: 0x2c400
  SPRAYER QUEUE: 32
  IPV4 MULTICAST: 0
  IPV6 MULTICAST: 0
  USE TABLE ID IPV4: 0
  USE TABLE ID IPV6: 0
  USE TABLE ID MPLS: 0
  TABLE ID: 0
  QOS ENABLE: 0
  QOS ID: 0
  NETFLOW SAMPLING PERIOD: 0
  L2 PKT DROP: 0
  L2 QOS ENABLE: 0
  SRC FWDING: 0
  *BUNDLE IFHANDLE: 0
```

```
  *TUNNEL IFHANDLE: 0
  *L2 ENCAP: 3

  * Not programmed in hardware
.
.
.
```