Configuring Enhanced Drop Detection and Enhanced Packet Drop Analyzer

Restrictions for Configuring Enhanced Drop Detection and Enhanced Packet Drop Analyzer

The following restrictions are applicable to Enhanced Drop Detection:

- All the traps are not documented in the list of traps for Enhanced Drop Detection.
- Traffic Manager traps need to be cleared after use to ensure the accuracy of the trap counters.

The following restrictions are applicable to Enhanced Packet Drop Analyzer:

- Traffic Manager traps that are enabled in Enhanced Packet Drop Analyzer display aggregated counters.
- In Enhanced Packet Drop Analyzer, the rate of packet capture is limited to 100 packets per second.
- Network Processing Unit traps need to be enabled selectively. There is no mechanism to enable all Network Processing Unit traps at once.

Information About Enhanced Drop Detection and Enhanced Packet Drop Analyzer

The following sections provide information about configuring Enhanced Drop Detection and Enhanced Packet Drop Analyzer.

Enhanced Drop Detection

Enhanced Drop Detection (EDD) extends the functionality of packet capture by allowing you to determine where packets are being dropped in the processing path. Packets can be dropped for various reasons. Enhanced Drop Detection helps in detecting the dropped packets and displays them. Packets are displayed in a tabular format with different counters that show the name of the packet, the previous counter value, the current counter value and the calculated delta value. EDD allows you to display all the packets dropped at an ASIC level. It can also display packets specifically dropped by the Network Processing Unit (NPU) or the Traffic Manager (TM).

EDD can also display the traps at an ASIC level. A trap is a mechanism that is used by the data path to raise events called trap events. Trap events are based on various conditions that are evaluated while processing a packet. Based on trap events a packet can be duplicated, dropped or redirected to a new destination. EDD can display NPU traps and TM traps. The types of traps which are displayed are drop traps, punt traps and snoop traps.

The information about the traps can be used to troubleshoot where the packets are being dropped. The list of traps in EDD displays all the traps, not just the traps that represent dropped packets. By displaying the traps at regular intervals you can identify the traps that have the largest changes in their delta values.

Using EDD and information from the traps can be the first step in troubleshooting where packets are being dropped in the processing path. Further, by using Enhanced Packet Drop Analyzer the nature and causes of the packet drops can be evaluated.

Enhanced Packet Drop Analyzer

Enhanced Packet Drop Analyzer (EPDA) allows you to select and configure traps to punt dropped packets to a CPU based destination. Traps are mechanisms that are created by the system to analyze different conditions during packet processing. Based on specific conditions traps can punt packets to different destinations.

EPDA allows you to configure specific Network Processing Unit (NPU) traps or Traffic Manager (TM) traps to punt dropped packets. All the TM traps can be enabled at once. But NPU traps need to be enabled one by one. Once EPDA is stopped, you should clear all traps to ensure they continue to work efficiently.

EPDA allows you to configure a buffer to capture the dropped packets. EPDA can display the details of the dropped packets and the statistics for all the dropped packets. This information can help in troubleshooting the reasons for packets being dropped.

Troublehsooting Packet Drop Using Enhanced Drop Detection and Enhanced Packet Drop Analyzer

EDD counts packets for all traps, including drop traps. EPDA allows you to configure specific drop traps to know which packets are being dropped. Using EDD along with EPDA you can get a greater understanding of why particluar packets are being dropped.

EDD displays all the traps. The previous, current and delta sections in the command outputs can help identify the relevant traps. Using the command multiple times will help you identify the traps with the largest changes in their outputs.

Once the relevant traps are identified, you can use EPDA to configure the identified traps to punt the dropped packets to the specified CPU destination. After the dropped packets have been punted, EPDA can display the details of the dropped packets. These details will help you understand the reasons for the packet drops.

After troubleshooting you need to clear all the traps. This is important to ensure the proper functionality of the traps.

Configuring Enhanced Drop Detection and Enhanced Packet Drop Analyzer to Troubleshoot Packet Drops

To troubleshoot where the packets are being dropped and the reasons for packet drops, perform this procedure

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**show platform hardware fed active fwd-asic traps** [**npu traps** | **tm traps**] **asic** [*asic instance* | **all**] **Example:** `Device# show platform hardware fed active fwd-asic traps npu-traps asic all` or `Device# show platform hardware fed active fwd-asic traps tm-traps asic 0`	Displays all the traps. The **npu-traps** keyword has both the *asic instance* and **all** options. The **tm-traps** keyword has only the *asic instance* option. Using the command multiple times can identify the traps with largest changes in their counters.
Step 3	**debug platform software fed active drop-capture set-trap** [**npu-traps** | **tm-traps**] *trap-name* **Example:** `Device# debug platform software fed active drop-capture set-trap npu-traps l3 l3-null-adj`	Configures the specified trap to punt dropped packets.

	Command or Action	Purpose
Step 4	**debug platform software fed active drop-capture start** **Example:** `Device# debug platform software fed active drop-capture start`	Starts the punting of dropped packets to the specified destination.
Step 5	**debug platform software fed active drop-capture stop** **Example:** `Device# debug platform software fed active drop-capture stop`	Stops the punting of dropped packets.
Step 6	**show platform software fed active drop packet-capture** **Example:** `Device# show platform software fed active drop packet-capture brief`	Displays the details of the dropped packets.
Step 7	**debug platform software fed active drop-capture clear-trap** **Example:** `Device# debug platform software fed active drop-capture clear-trap npu-traps l3 l3-null-adj`	Clears the traps configured to punt dropped packets.

Configuration Examples for Enhanced Drop Detection and Enhanced Packet Drop Analyzer

Example: Configuring Enhanced Drop Detection

The following examples shows how to display all the packets dropped at an ASIC level:

```
Device# sh platform hardware fed active fwd-asic drops asic 0
Note: Slice and IFG showing -1 are global counters
```

```
|  #     |                      Counters Name                          |slice_number|
ifg_number  |  prev_value   |  current_value  |     delta      |
=============================================================================================
|  1     |Fwd drop counter (DSP==1): pkts                              |          -1|
      -1|              0|                0|              0|
|  2     |Fwd drop counter (DSP==1): bytes                             |          -1|
      -1|              0|                0|              0|
|  3     |LM drop counter for slice                                    |          -1|
      -1|              0|                0|              0|
|  4     |RX IFGB6 Port0 drop_counter TC0                              |           3|
       0|              0|                0|              0|
|  5     |RX IFGB6 Port0 drop_counter TC1                              |           3|
```

```
       0|              0|                0|               0|
|  6    |RX IFGB6 Port0 drop_counter TC2                                |           3|
       0|              0|                0|               0|
|  7    |RX IFGB6 Port0 drop_counter TC3                                |           3|
       0|              0|                0|               0|
|  8    |RX IFGB6 Port4 drop_counter TC0                                |           3|
       0|              0|                0|               0|
|  9    |RX IFGB6 Port4 drop_counter TC1                                |           3|
       0|              0|                0|               0|
|  10   |RX IFGB6 Port4 drop_counter TC2                                |           3|
       0|              0|                0|               0|
```

The following example shows how to display all the NPU traps at an ASIC level:

```
Device# show platform hardware fed active fwd-asic traps npu-traps asic all
Trap ID |     Asic     |                    NPU Trap Name                    |     Prev
 |   Current    |    Delta
-------------------------------------------------------------------------------------------------------------

     1 |            0 |                         la_event_e_ETHERNET_ACL_DROP |            0
 |            0 |            0
     2 |            0 |                  la_event_e_ETHERNET_ACL_FORCE_PUNT |            0
 |            0 |            0
     4 |            0 |       la_event_e_ETHERNET_NO_TERMINATION_ON_L3_PORT |            0
 |            0 |            0
     5 |            0 |                 la_event_e_ETHERNET_CISCO_PROTOCOLS |            2
 |            2 |            0
     6 |            0 |                        la_event_e_ETHERNET_DA_ERROR |            0
 |            0 |            0
     7 |            0 |                    la_event_e_ETHERNET_DHCPV4_CLIENT |            0
 |            0 |            0
     8 |            0 |                    la_event_e_ETHERNET_DHCPV4_SERVER |            0
 |            0 |            0
     9 |            0 |                    la_event_e_ETHERNET_DHCPV6_CLIENT |            0
 |            0 |            0
   <snip>
```

The following example shows how to display all the TM traps at an ASIC level:

```
Device# show platform hardware fed active fwd-asic traps tm-traps asic all
Warning:
1. Per VOQ Counters will be affected, Please disable TM Traps in EDD CLI using clear option
2. Please note TM traps enabled in EPDA will display aggregated counters
-------------------------------------------------------------------------------------------------------------

Trap ID | Asic |                                      TM Trap
      |     Prev     |   Current    |    Delta
-------------------------------------------------------------------------------------------------------------

 1 |  0 |                la_tm_traps_e_EXACT_METER_PACKET_GOT_DROPPED_DUE_TO_EXACT_METER |
           0 |            0 |            0
 2 |  0 |    la_tm_traps_e_STATISTICAL_METER_PACKET_GOT_DROPPED_DUE_TO_STATISTICAL_METER |
           0 |            0 |            0
 3 |  0 |                                la_tm_traps_e_ETHERNET_METERS_PACKET_OUT_OF_RATE |
           0 |            0 |            0
 4 |  0 |                                     la_tm_traps_e_RXPDR_A_COUNTER_DSP_OVERFLOW |
           0 |            0 |            0
 5 |  0 |                                      la_tm_traps_e_RXPDR_A_COUNTER_VN_OVERFLOW |
           0 |            0 |            0
 6 |  0 |                                    la_tm_traps_e_RXPDR_A_COUNTER_MCID_OVERFLOW |
           0 |            0 |            0
 7 |  0 |                                         la_tm_traps_e_RXPDR_B_COUNTER_OVERFLOW |
           0 |            0 |            0   <snip>
```

Example: Configuring Enhanced Packet Drop Analyzer

The following example shows how to enable NPU traps. NPU traps need to be enable one at a time:

```
Device# #debug platform software fed active drop-capture set-trap npu-traps ?
  L3        npu layer 3 traps
  OAMP      npu OAMP traps
  app       npu app traps
  ethernet  npu ethernet traps
  internal  npu internal traps
  ipv4      npu ipv4 traps
  ipv6      npu ipv6 traps
  mpls      npu mpls traps

Device# debug platform software fed active drop-capture set-trap npu-traps l3 ?
l3-absr-tbl-miss           npu trap l3 absr table miss
  l3-bfd-mic-ip-dis          npu trap l3 bfd micro ip disabled
  l3-drop-adjacency          npu trap l3 drop adj
  l3-enc-tbl-miss            npu trap l3 enacap table miss
  l3-int-hop-limit           npu trap l3 int hop limit
  l3-invalid-spi             npu trap l3 invalid spi
  l3-ip-mc-drop              npu trap l3 ip mc drop
  l3-lpm-def-drop            npu trap l3 lpm default drop
<snip>
```

The following exampl shows how to enable TM traps. All the TM traps can be enabled at once:

```
Device# #debug platform software fed active drop-capture set-trap tm-traps ?
  all                           Enable all TM Traps
  eth-meter-oor                 Ethernet Meter packets Out of rate
  exact-meter                   Exact Meter Drop reason
  rxpdr-b-overflow              RXPDR B overflow
  rxpdr-dsp-overflow            RXPDR DSP overflow
<snip>
Device# #debug platform software fed active drop-capture set-trap tm-traps all
```

The following example shows how to start punting the dropped packets:

```
Device# debug platform software fed active drop-capture start
```

The following example shows how to stop punting the dropped packets:

```
Device# debug platform software fed active drop-capture stop
```

The following example shows how to display the details of the dropped packets in brief:

```
Device# show platform software fed active drop packet-capture brief
DropPackets packet capturing: disabled. Buffer wrapping: disabled
Total captured so far :  2313 packet(s)
Capture capacity      :  4096 packet(s)
Max. Meta header size :    88 byte(s)
Max. Packet data size :   128 byte(s)

------ DropPackets Packet Number: 1, Timestamp: 2024/03/25 15:04:46.823 ------
 interface : phy:  [if-id: 0x00000000], pal:  [if-id: 0x00000000]
 misc info : cause: 0 [Reserved ], sub-cause: 0, linktype: UNKNOWN [0]
 CE    hdr : dest mac: 4e41.5000.0111, src mac: 4e41.5000.0111, ethertype: 0x7106
 meta  hdr : Nxt. Hdr: 0x1, Fwd. Hdr: 0x2, SSP: 0x19
 meta  hdr : DSP: 0xffff, SLP: 0xe, DLP: 0x95
 ether hdr : dest mac: 341b.2d76.fd02, src mac: 6c29.d29d.36c3
 ether hdr : vlan: 3012, ethertype: 0x8100
 ipv4  hdr : dest ip: 172.16.10.11, src ip: 192.168.100.18
 ipv4  hdr : packet len: 100, ttl: 254, protocol: 1 (ICMP)
 icmp  hdr : icmp type: 8, code: 0
```

The following example shows how to display the details of the dropped packets in detail. The detailed view includes information about the source interface. This information is not present in the brief view:

```
Device# show platform software fed active drop packet-capture detailed
DropPackets packet capturing: disabled. Buffer wrapping: disabled
Total captured so far :     1 packet(s)
Capture capacity      :  4096 packet(s)
Max. Meta header size :    88 byte(s)
Max. Packet data size :   128 byte(s)

------ DropPackets Packet Number: 1, Timestamp: 2024/07/11 13:25:48.157 ------
 interface : phy:  [if-id: 0x00000000], pal:  [if-id: 0x00000000]
 misc info : cause: 0 [Reserved ], sub-cause: 0, linktype: UNKNOWN [0]
 CE    hdr : dest mac: 4e41.5000.0111, src mac: 4e41.5000.0111, ethertype: 0x7106
 meta  hdr : Nxt. Hdr: 0x1, Fwd. Hdr: 0, SSP: 0xffff
 meta  hdr : DSP: 0x1d, SLP: 0x8002a, DLP: 0x2a
 ether hdr : dest mac: 0900.2b00.0005, src mac: 6c29.d293.6f46
 ether hdr : length: 1402

 Metadata Hex-Dump (captured metadata length: 54 bytes) :
 4E41500001114E41  5000011171060800  0C3500FFFF001D80  02A0002A0064000A
 74B91E3B00751E3A  F821100200400000  000000000000

 Packet Data Hex-Dump (captured packet length: 128 bytes) :
 09002B0000056C29  D2936F46057AFEFE  0383140100110100  0002010098004013
 001E057725D30300  0000F00502000000  258101CC01040349  00008404C0A828B2
 08FF000000000000  0000000000000000  0000000000000000  0000000000000000
 0000000000000000  0000000000000000  0000000000000000  0000000000000000

 Punt Header
 Nxt. Hdr        = 0x1     Ethernet         Fwd. Hdr        = 0       Ethernet
 Punt Padding    = 0                        Next Hdr. Off.  = 0
 Source         = 0xc     Eg. Trap        Code           = 0x35    ETHERNET_SAME_INTERFACE

 Flow Type       = 0                        SSP             = 0xffff
 DSP             = 0x1d    HundredGigE1/0/5  SLP             = 0x8002a Port5lVlan100
 DLP             = 0x2a                     Padding         = 0
 Relay ID        = 0x64                     Time Stamp      = 0xa74b91e3b0075

 Receive Time    = 0x1e3af821
```

The following example shows how to clear all traps:

```
Device# debug platform software fed active drop-capture clear-trap npu-traps all
```

or

```
Device# debug platform software fed active drop-capture clear-trap tm-traps all
```

Note All TM traps need to be cleared after each use.

Feature History for Enhanced Drop Detection and Enhanced Packet Drop Analyzer

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Dublin 17.12.4 This feature is not applicable to the Cisco IOS XE 17.13.x release and the Cisco IOS XE 17.14.x release.	Enhanced Drop Detection	Enhanced Drop Detection allows you to determine where packets are being dropped in the processing path. Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).
Cisco IOS XE Dublin 17.12.4 This feature is not applicable to the Cisco IOS XE 17.13.x release and the Cisco IOS XE 17.14.x release.	Enhanced Packet Drop Analyzer	Enhanced Packet Drop Analyzer allows you to configure traps to punt dropped packets to a CPU based destination for the purpose of troubleshooting. Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.”