Troubleshoot Input Drops in Cisco IOS XR

Contents

Introduction
Prerequisites
Requirements
Components Used
Background Information
Problem: Increment in the Input Drop
Controller Drops
Unknown DestinationMedium Access Control Address (DMAC) or dot1q VLAN
Packets Dropped Due to Unrecognized Upper-Level Protocol
NP Drops on the ASR 9000
Netio

Introduction

This document describes how to troubleshoot input drops on the interface on XR routers.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This article covers ASR 9000 series routers, CRS series routers, and GSR 12000 series routers.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Input drops in Cisco IOS XR have a completely different meaning than that of input drops in Cisco IOS. It can confuse you when it migrates Cisco IOS to Cisco IOS XR and starts to see its input drop counters in the show interface.

In Cisco IOS, an input drop was due to the interface input queue that gets full. This means that too many packets were punted to the CPU for process switching and it was not able to handle them fast enough. The input queue is built up until it gets full and there are some drops.

In Cisco IOS XR, there is no strict definition of an input drop. So it is basically up to the developers of a component to decide whether they want to increment the input drop counter when they decide to drop a packet. The key thing here is that at some point in the code, the router decides to drop the packet, so that means that it is likely that router was not supposed to forward that packet, and router decided consciously to drop it. Therefore, this is not related to congestion like in Cisco IOS. However, it is rather a packet which was received by the router and which was not supposed to be forwarded, so router decided to drop it and it is very likely not a reason to be alarmed. Although, you cannot tell whether it is something to worry about or not until you have completely understood the kind of packets that are incrementing the input drop counter, and that is not that simple.

Examples:

- An XR router is connected to a switch which sends some bridge protocol data unit (BPDUs) andUDLD packets. The XR router does not have spanning-tree nor UDLD configured on its layer 3 interfaces so it just drops these frames and it increments the input drops counter in show interface. In this case, there is nothing to worry about as it is the right thing to do to drop these frames as the features are not configured.
- An ASR 9000 has a Cisco Express Forwarding (CEF) entry wrongly programmed due to a bug so that it does not point at a valid adjacency. In this case, the Network Processor of the ASR 9000 Line Card (LC) realizes that router misses a load info, and increments a Network Processor (NP) drop counter which is uploaded to the interface input drop counter.

When the input drops are reported, the problem is to figure out whether these are legitimate drops like in example 1, or the consequence of a problem like in example 2.

Problem: Increment in the Input Drop

This document enlists the reasons for the input drops that are incremented and how to check if it is that reason:

Controller Drops

Runts, Frame Check Sequence (FCS), aborts, First Input First Output (FIFO) overflows, giants Packet Over SDH/SONET (POS) drops.

```
RP/0/RP0/CPU0:equinox#show controllers poS 0/2/0/0 framer statistics
POS Driver Internal Cooked Stats Values for port 0
===================================================
Rx Statistics                   Tx Statistics
-------------                   -------------
Total Bytes:    71346296        Total Bytes:    67718333
Good Bytes:     71346296        Good Bytes:     67718333
Good Packets:   105385          Good Packets:   67281
Aborts:         0               Aborts:         0
FCS Errors:     0               Min-len errors: 0
Runts:          0               Max-len errors: 0
FIFO Overflows: 0               FIFO Underruns: 0
Giants:         0
Drops:          0

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

For an ethernet (gige, tengige...) interface, check something like:

```
show controllers gigabitEthernet 0/0/0/18 stats
```

See if there is one counter in the controller stats which increments at the same rate as the input drop counter in show interface. Some of these error counters must also be in show interface.

Unknown Destination Medium Access Control Address (DMAC) or dot1q VLAN

Packets with a destination MAC address, not the one of the interface, or with a Virtual Local Area Network (VLAN) not matched by a subinterface. These can happen when there is some flooding in an L2 domain of unknown unicast MAC addresses so the XR router connected to that L2 domain receives frames with a destination MAC address which is not one of its controllers. It is also possible when an Cisco IOS router is sending ethernet keepalives on its gige interface, so these keepalives are incrementing the input drops on the XR router as they do not have the destination mac address of the XR router. Also, when the interface is connected to another device which has more dot1q vlans/subinterfaces configured as on the XR router so that the XR router receives frames with an unknown dot1q tag.

On a CRS fixed Physical Layer Interface Modules (PLIM), you could find such drops in:

<#root>

```
RP/0/RP0/CPU0:pixies-uk#sh contr plim asic statistics interface tenGigE 0/1/0/3 location 0/1/CPU0
Wed Aug 22 16:07:47.854 CEST
                               Node: 0/1/CPU0


TenGigE0/1/0/3 Drop
-------------------------------------------------------------------------------
RxFiFO Drop       : 0                   PAR Tail Drop     : 0
PAR Err Drop      : 0                   Invalid MAC Drop  : 86


TxFIFO Drop       : 0                   Invalid VLAN Drop : 11
```

Or in the tengige or gige controller stats:

```
RP/0/RP0/CPU0:pixies-uk#sh contr ten 0/1/0/3 stats
Wed Aug 22 16:22:42.059 CEST
Statistics for interface TenGigE0/1/0/3 (cached values):

Ingress:


    Input drop overrun          = 0
    Input drop abort            = 0
    Input drop invalid VLAN     = 11
    Input drop invalid DMAC     = 0
    Input drop invalid encap    = 0
    Input drop other            = 86
```

Note: Cisco bug ID CSCub74803 exist. Input drop other is incremented instead of Input drop invalid DMAC at least on the 8-port tengige fixed PLIM of the CRS.

For Shared Port Adapters (SPAs) (CRS, XR 12000), the packets with invalid MAC would be dropped by the SPA l2-tcam so you can find these drops in show controllers TenGigE a/b/c/d all:

```
    Input drop other            = 107

   l2-tcam Invalid DA Drops: 107
```

On an ASR 9000, the Input drop invalid DMAC and Input drop other counters in the controller stats are not incremented. So the way to recognize these drops on the ASR 9000 is to find the NP handling the interface with the input drops:

```
RP/0/RSP0/CPU0:obama#sh int gig 0/0/0/30 | i "input drops"
Wed Aug 22 16:55:52.374 CEST
     1155 packets input, 156256 bytes, 1000 total input drops
RP/0/RSP0/CPU0:obama#sh contr np ports all location 0/0/CPU0
Wed Aug 22 16:56:01.385 CEST

                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:obama#
```

You can see that the interface gig 0/0/0/30 is handled by the NP 0 on 0/0/CPU0.
Let us check the NP counters of NP0 on 0/0/CPU0:

<#root>

```
RP/0/RSP0/CPU0:obama#sh contr np counters np0 location 0/0/CPU0
Wed Aug 22 16:56:19.883 CEST

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

Show global stats counters for NP0, revision v3

Read 26 non-zero NP counters:
Offset  Counter                                         FrameValue   Rate (pps)
```

```
--------------------------------------------------------------------------------
  22  PARSE_ENET_RECEIVE_CNT                                    1465           0
  23  PARSE_FABRIC_RECEIVE_CNT                                  2793           0
  24  PARSE_LOOPBACK_RECEIVE_CNT                                2800           0
  28  MODIFY_FABRIC_TRANSMIT_CNT                                  80           0
  29  MODIFY_ENET_TRANSMIT_CNT                                  1792           0
  32  RESOLVE_INGRESS_DROP_CNT                                  1000           0
  35  MODIFY_EGRESS_DROP_CNT                                    1400           0
  36  MODIFY_MCAST_FLD_LOOPBACK_CNT                             1400           0
  38  PARSE_INGRESS_PUNT_CNT                                     465           0
  39  PARSE_EGRESS_PUNT_CNT                                      155           0
  45  MODIFY_RPF_FAIL_DROP_CNT                                  1400           0
  53  PARSE_LC_INJECT_TO_FAB_CNT                                  80           0
  54  PARSE_LC_INJECT_TO_PORT_CNT                                864           0
  57  PARSE_FAB_INJECT_UNKN_CNT                                  155           0
  67  RESOLVE_INGRESS_L3_PUNT_CNT                                465           0
  69  RESOLVE_INGRESS_L2_PUNT_CNT                                464           0
  70  RESOLVE_EGRESS_L3_PUNT_CNT                                1400           0
  93  CDP                                                        464           0
  95  ARP                                                          1           0
 109  DIAGS                                                      154           0
 221  PUNT_STATISTICS                                           9142           1
 223  PUNT_DIAGS_RSP_ACT                                         155           0
 225  PUNT_DIAGS_RSP_STBY                                        155           0
 227  NETIO_RP_TO_LC_CPU_PUNT                                    155           0


373  L3_NOT_MYMAC                                              1000           0


 565  INJECT_EGR_PARSE_PRRT_PIT                                  928           0

RP/0/RSP0/CPU0:obama#
```

So L3_NOT_MYMAC in the NP counter means that the router received a frame on a Layer 3 interface with a destination MAC address which is not one of the interfaces. The router drops it as expected, and this is reported as input drops in show interface.

On the ASR 9000 for packets received with a dot1q VLAN not configured on a subinterface of the ASR 9000, the Input drop unknown 802.1Q counter is not incremented in show controllers gigabitEthernet 0/0/0/30 stats. The procedure is the same as above for the unknown DMAC: identify which NP handles the interfaces, and then check this NP counter. You see that the NP counter UIDB_TCAM_MISS_AGG_DROP incrementing in that case.

Packets Dropped Due to Unrecognized Upper-Level Protocol

That one is straightforward to detect as there is a counter for these drops one line below the input drops in show interface:

```
RP/0/RSP0/CPU0:obama#sh int gig 0/0/0/18
Wed Aug 22 17:14:35.232 CEST
GigabitEthernet0/0/0/18 is up, line protocol is up
```

```
  5 minute input rate 4000 bits/sec, 0 packets/sec
  5 minute output rate 5000 bits/sec, 0 packets/sec
     7375 packets input, 6565506 bytes, 1481 total input drops
     1481 drops for unrecognized upper-level protocol
```

You can see here that all input drops were due to unrecognized upper-level protocol.

That means that packets were received with an Ethernet protocol that router is not interested in. That means that a feature is configured on the neighbor (or on a host connected to the layer 2 domain connected to that interface) so that it sends you frames with a protocol not configured on the XR router.

Examples: BPDUs, Intermediate System to Intermediate System (ISIS), Connection Less Network Protocol (CLNP), IPv6, UDLD, Cisco Discovery Protocol (CDP), VLAN Trunking Protocol (VTP), Dynamic Trunking Protocol (DTP), Link Layer Discovery Protocol (LLDP) etc....

When these features are not configured on the XR interface, then the XR box drops them as expected. To find out what kind of frames are incrementing this counter, you can have to compare which features are enabled on the XR router with the features enabled on the neighbor (it can be a router or a switch), or the features enabled on all the devices connected to the layer 2 domains connected to that interface (much less easy). If the XR router is connected to a switch, you can try a span on that switch of the packets that it sends to the XR router on the interface with input drops.
ASR9000/XR : Drops for Unrecognized Upper-Level Protocol Error

NP Drops on the ASR 9000

Drops counters in the Network Process (NP) of the ASR 9000 are reported as input drops when they apply to a packet received on an interface and dropped. This does not happen for Packet Switch Engine (PSE) drops on the CRS and the XR 12000. They are not counted as input drops.

So if you have input drops on an ASR 9000 and they do not match one of these reasons, then you would do a **show controllers np ports all location 0/<x>/CPU0** to find the NP handling the interface with input drops and then check its NP counters with **show contr np counters np<y> location 0/<x>/CPU0**.

You can pipe the output to only keep DROP counters with a command like **sh contr np counters np<y> location 0/<x>/CPU0 | i DROP**, but this can be dangerous as sometimes a drop counter does not have DROP in its name. You have seen a good example with L3_NOT_MYMAC. So maybe pipe for **DROP|DISCARD|NOT|EXCD**.

You can clear the interface counters and the NP counters with **clear controller np counters np<y> location 0/<x>/CPU0** at roughly the same time to find out which NP counter increments at the same rate as the input drops.

For example, you get IPV4_PLU_DROP_PKT in the NP counters which means that the CEF/PLU entry is saying that the packet has to be dropped. You do not have a default route and have unreachables disabled so packets not matching a more specific route, where hitting the default CEF handler, is a drop entry.

If you find a drop counter in the NP which can explain the input drops as they increment at the same rate, but the NP drop counter is not very self-explanatory, you can navigate this page to try to understand what the counter means:
ASR9000/XR: Troubleshoot Packet Drops and Understanding NP Drop Counters

Note: Xander's page on support forums contains the drop reasons for the first generation of linecards (Trident), and there are new counter names for the new generation (Typhoon) linecards. Based on the name, you must be able to find a similar counter name as on Trident.

Netio

You can collect a **show netio idb <int>** and this gives you the interface input drop and the netio node drop counters:

<#root>

```
RP/0/RP0/CPU0:ipc-lsp690-r-ca-01#show netio idb gigabitEthernet 0/2/0/1

GigabitEthernet0/2/0/1 (handle: 0x01280040, nodeid:0x21) netio idb:
---------------------------------
name:                    GigabitEthernet0_2_0_1
interface handle:        0x01280040
interface global index:  3
physical media type:     30
dchain ptr:              <0x482e0700>
echain ptr:              <0x482e1024>
fchain ptr:              <0x482e13ec>
driver cookie:           <0x4829fc6c>
driver func:             <0x4829f040>
number of subinterfaces: 4096
subblock array size:     7
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`

```
 <=========== cleared when added to input drop counter !!!
   OUT drop pkts:           0
   IN  errors pkts:         0
   OUT errors pkts:         0

Chains
--------------------
Base decap chain:
     ether                <30>  <0xfd018cd8, 0x482c736c>  <       0,         0>

Protocol chains:
---------------
<Protocol number> (name)  Stats
  Type  Chain_node       <caps num>  <function, context> <drop pkts, drop bytes>
  <snip>
  <13> (mpls)   Stats IN: 204 pkts, 23256 bytes; OUT: 0 pkts, 0 bytes
    Encap:
     mpls                 <25>  <0xfcc7ddbc, 0x00000000>  <       0,         0>
     ether                <30>  <0xfd0189b4, 0x482c736c>  <       0,         0>
     l2_adj_rewrite       <86>  <0xfcaa997c, 0x4831a2e8>  <       0,         0>
     pcn_output           <54>  <0xfd0561f0, 0x48319f04>  <       0,         0>
     q_fq                 <43>  <0xfd05f4b8, 0x48320fec>  <       0,         0>
     txm_nopull          <60>  <0xfcadba38, 0x4824c0fc>  <       0,        0>
    Decap:
     pcn_input           <55>  <0xfd0561f0, 0x4830ba8c>  <       0,        0>
     q_fq_input          <96>  <0xfd05f330, 0x48312c7c>  <       0,        0>

     mpls                <25>  <0xfcc7b2b8, 0x00000000>  <     152,    17328>


    Fixup:
     l2_adj_rewrite      <86>  <0xfcaa945c, 0x00000000>  <       0,        0>
     pcn_output          <54>  <0xfd0561f0, 0x48319f04>  <       0,        0>
     q_fq                <43>  <0xfd05f4b8, 0x48320fec>  <       0,        0>
     txm_nopull          <60>  <0xfcadba38, 0x4824c0fc>  <       0,        0>
```

The drops in the Multi-Protocol Label Switching (MPLS) node here could be due to MPLS Time To Live (TTL) expired (in case of a loop or when the customer does a traceroute), or fragmentation required and Do Not Fragment (DF) bit set for instance. You can run **debug mpls packet drop** and **debug mpls error** with the location of the interface to try to figure out what kind of packet is incrementing this counter.

Punted multicast packets. When you see netio IN drop pkts but no netio node below with some drops which could explain the IN drop pkts, then this can be mcast punted packets, and you can enable deb mfib netio drop to figure out what kind of packets