

排除Cisco IOS XR中的输入丢弃故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[问题：输入丢弃中的增量](#)

[控制器丢弃](#)

[未知目标介质访问控制地址\(DMAC\)或dot1q VLAN](#)

[由于无法识别的上层协议而丢弃的数据包](#)

[ASR 9000上的NP丢弃](#)

[内蒂奥](#)

简介

本文档介绍如何对XR路由器的接口上的输入丢弃进行故障排除。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文介绍ASR 9000系列路由器、CRS系列路由器和GSR 12000系列路由器。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

Cisco IOS XR中的输入丢弃与Cisco IOS中的输入丢弃具有完全不同的含义。当它将Cisco IOS迁移到Cisco IOS XR并开始在其show interface中看到其输入丢弃计数器时，可能会让您感到困惑。

在Cisco IOS中，输入丢弃是由于接口输入队列已满引起的。这意味着有太多数据包被传送到CPU进行进程交换，而且处理速度不够快。输入队列将建立到已满并且出现一些丢包为止。

在Cisco IOS XR中，输入丢弃没有严格的定义。因此，基本上由组件的开发人员决定，当他们决定丢弃数据包时，他们是否要增加输入丢弃计数器。这里的关键点是，在代码中的某个时刻，路由器

决定丢弃数据包，这意味着路由器很可能不应该转发该数据包，而路由器决定有意识地丢弃该数据包。因此，这与Cisco IOS中的拥塞无关。但是，它实际上是一个由路由器接收且不应转发的数据包，因此路由器决定丢弃该数据包，因此很可能不会引起警报。不过，在完全了解了使输入丢弃计数器递增的数据包类型之前，您无法判断它是否值得担忧，但这不是那么简单。

示例：

- XR路由器连接到交换机，该交换机发送一些网桥协议数据单元(BPDU)和UDLD数据包。XR路由器的第3层接口上未配置生成树和UDLD，因此它只丢弃这些帧，并在show interface中增加输入丢弃计数器。在这种情况下，无需担心任何问题，因为在没有配置功能的情况下，丢弃这些帧是正确之举。
- 由于Bug，ASR 9000的思科快速转发(CEF)条目被错误编程，因此它不会指向有效的邻接关系。在这种情况下，ASR 9000线路卡(LC)的网络处理器会发现路由器遗漏了负载信息，并会增加网络处理器(NP)丢弃计数器，该计数器会上传到接口输入丢弃计数器。

当报告输入丢弃时，问题是要弄清楚它们是合法的丢弃（如示例1所示）还是问题的后果（如示例2中）。

问题：输入丢弃中的增量

本文档列出了导致输入丢弃增加的原因以及如何检查是否是原因：

控制器丢弃

残帧、帧校验序列(FCS)、中止、第一输入第一输出(FIFO)溢出、SDH/SONET上的超大型数据包(POS)丢弃。

```
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#
```

对于以太网(gige、tengige...)接口，请检查类似以下内容：

show controllers gigabitEthernet 0/0/0/18统计信息

查看控制器统计信息中是否有与show interface中的输入丢弃计数器以相同速率递增的计数器。其中某些错误计数器也必须位于show interface中。

未知目标介质访问控制地址(DMAC)或dot1q VLAN

具有目的MAC地址（不是接口地址）或具有子接口不匹配的虚拟局域网(VLAN)的数据包。当未知单播MAC地址的L2域中有一些泛洪时，就会发生这种情况，因此连接到该L2域的XR路由器收到包含目标MAC地址的帧，而目的MAC地址不是其控制器之一。当Cisco IOS路由器在其gige接口上发送以太网keepalive时，也有可能发生，因此这些keepalive将增加XR路由器上的输入丢弃，因为它们没有XR路由器的目标mac地址。此外，当接口连接到在XR路由器上配置了更多dot1q vlan/子接口的另一台设备时，使XR路由器接收带有未知dot1q标记的帧。

在CRS固定物理层接口模块(PLIM)上，您可以在以下位置找到此类丢包：

<#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
```


或者，在tengige或gige控制器统计信息中：

```
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
```

 注意：Cisco Bug ID [CSCub74803](#)存在。至少在CRS的8端口固定的PLIM上，Input drop other会递增而不是输入丢弃无效DMAC。

对于共享端口适配器(SPA)(CRS、XR 12000),SPA I2-tcam将丢弃具有无效MAC的数据包，因此您可以在show controllers TenGigE a/b/c/d all中找到这些丢包：

```
Input drop other              = 107
```

```
I2-tcam Invalid DA Drops: 107
```

在ASR 9000上，控制器统计信息中的Input drop invalid DMAC和Input drop other counters不会增加。因此，在ASR 9000上识别这些丢弃的方法是找到使用输入丢弃处理接口的NP:

```
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#
```

您可以看到接口gig 0/0/0/30由0/0/CPU0上的NP 0处理。
让我们检查0/0/CPU0上NP0的NP计数器：

```
<#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#

因此，NP计数器中的L3_NOT_MYMAC意味着路由器在第3层接口上收到一个帧，其目的MAC地址不是该接口之一。路由器按预期丢弃它，这在show interface中报告为输入丢弃。

在ASR 9000上，对于ASR 9000的子接口上未配置dot1q VLAN接收的数据包，Input drop unknown 802.1Q计数器不会在show controllers gigabitEthernet 0/0/0/30统计信息中递增。未知DMAC的步骤与上面相同：确定哪个NP处理接口，然后检查此NP计数器。在这种情况下，您会看到NP计数器UIDB_TCAM_MISS_AGG_DROP递增。

由于无法识别的上层协议而丢弃的数据包

该计数器很容易检测，因为show interface中的输入丢弃下面有一行用于这些丢弃的计数器：

```
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
```

您可以在这里看到，所有输入丢弃都是由于无法识别的上层协议造成的。

这意味着收到的数据包使用的是路由器不感兴趣的以太网协议。这意味着在邻居（或连接到该接口的第2层域的主机）上配置了一个功能，这样它就可以使用未在XR路由器上配置的协议发送帧。

示例：BPDU、中间系统到中间系统(ISIS)、无连接网络协议(CLNP)、IPv6、UDLD、思科发现协议(CDP)、VLAN中继协议(VTP)、动态中继协议(DTP)、链路层发现协议(LLDP)等....

如果未在XR接口上配置这些功能，则XR框会按预期丢弃它们。要了解哪种帧会增加此计数器，您必须将XR路由器上启用的功能与邻居上启用的功能（可以是路由器或交换机）进行比较，或者比较连接到与该接口相连的第2层域的所有设备上启用的功能（简单得多）。如果XR路由器连接到交换机，您可以在该交换机上尝试使用输入丢弃在接口上发送到XR路由器的数据包的跨度。

[ASR9000/XR：因无法识别的上层协议错误而丢弃](#)

ASR 9000上的NP丢弃

当ASR 9000的网络进程(NP)中的丢弃计数器应用于接口上收到并丢弃的数据包时，它们被报告为输入丢弃。在CRS和XR上数据包交换引擎(PSE)丢弃不会发生此情况。它们不计为输入丢弃。


因此，如果您在ASR 9000上存在输入丢弃，并且它们不符合以下原因之一，则您将执行show controllers np ports all location 0/<x>/CPU0以查找处理输入丢弃接口的NP，然后使用show controllers np counters np<y> location 0/<x>/CPU0检查其NP计数器。

您可以使用命令(如sh contrr np counters np<y> location 0/<x>/CPU0)传输输出以仅保留DROP计数器 | i DROP，但有时由于丢弃计数器的名称中没有DROP，因此这种操作非常危险。您已经看到一个使用L3_NOT_MYMAC的很好示例。因此，可能为DROP|DISCARD|NOT|EXCD建立管道。

您可以使用几乎同时清除接口计数器和NP计数器和clear controller np counters np<y> location 0/<x>/CPU0，以找出哪个NP计数器以与输入丢弃相同的速率增加。

例如，您在NP计数器中收到IPV4_PLU_DROP_PKT，这意味着CEF/PLU条目表示必须丢弃数据包。您没有默认路由并禁用了不可达，因此不匹配更具体的路由（点击默认CEF处理程序时）的数据包是丢弃条目。

如果您在NP中找到一个丢弃计数器，它可以解释以相同速率递增的输入丢弃，但NP丢弃计数器并不是非常容易解释的，您可以浏览此页面来尝试理解计数器的含义：

 注意：Xander在支持论坛上的页面包含第一代线卡(Trident)的丢弃原因，并且新一代(Typhoon)线卡有新的计数器名称。根据名称，您必须能够找到与Trident中类似的计数器名称。

内蒂奥

您可以收集show netio idb <int>，这会为您提供接口输入丢弃和netio节点丢弃计数器：

<#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  
DSNCF: 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>
```

```

12_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:
12_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>

```

此处的多协议标签交换(MPLS)节点中的丢弃可能是由于MPLS生存时间(TTL)过期 (在出现环路或客户执行traceroute时) , 或者需要分段并且设置了Do Not Fragment(DF)位。您可以对接口的位置运行debug mpls packet drop和debug mpls error , 以尝试了解哪种数据包会增加此计数器。

传送组播数据包。当您看到netio IN丢弃数据包但下面没有netio节点以及一些可以解释IN丢弃数据包的丢弃时, 这可以是组播转发数据包, 并且您可以启用deb mfib netio drop以确定数据包的类型

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。