Nexus 9000云扩展ASIC CRC识别和跟踪程序

目录

简介

本文档介绍用于跟踪在一系列Cisco Nexus 9000云扩展ASIC模块的物理接口上观察到的CRC错误来源的步骤。本文档还介绍了用于区分模块化Nexus交换机的物理接口和内部交换矩阵链路上观察到的堆叠和非堆叠的CRC错误的步骤。

先决条件

要求

Cisco建议您了解直通交换和存储转发交换的基础知识。Cisco还建议您了解以太网FCS（帧校验序列）字段的基础知识以及FCS字段使用的CRC（循环冗余校验）算法。有关详细信息，请参阅以下文档：

- 适用于低延迟环境的直通和存储转发以太网交换

使用的组件

本文档中的信息基于运行NX-OS软件版本7.0(3)I7(8)的Cisco Nexus 9000系列交换机以及云扩展ASIC。

本文档中的信息在特定实验室环境设备上创建。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

背景信息

默认情况下，Cisco Nexus 9000系列交换机使用直通交换。直通交换是指交换机对帧做出转发决策，并在交换机处理了足够的帧报头做出有效转发决策后，开始从出口接口转发帧。这与存储转发交换不同，存储转发交换是指交换机在将帧从出口接口转发出来之前缓冲整个帧。

以太网帧的FCS字段验证帧的完整性，并确保帧在传输过程中未损坏。以太网帧的FCS字段位于该帧负载后的以太网帧的末尾。在存储转发交换模式下运行的交换机能够在将帧从出口接口转发出去（或如果FCS字段包含无效内容，则丢弃该帧）之前使用FCS字段验证以太网帧的完整性。 但是，在直通交换模式下运行的交换机无法在将以太网帧从出口接口转发出去之前使用FCS字段验证其完整性；换句话说，当直通交换机能够验证以太网帧的完整性时，大多数以太网帧已经从出口接口转发出去。

如果在直通交换模式下运行的交换机收到带有无效FCS字段的以太网帧，该交换机将执行以下操作：

1. 使用当前（不正确）FCS字段值的位逆值重写以太网帧的FCS字段。如果需要路由该帧，则在重写该帧的以太网报头后，将计算当前（不正确）的FCS字段值。此操作称为“堆叠”CRC。
2. 根据对以太网帧的转发决策，将以太网帧的剩余部分（以及存储的CRC）从出口接口转发出去。
3. 增加入口接口上的输入错误计数器和/或CRC错误计数器。

本文档介绍验证与入口接口关联的CRC计数器是普通CRC（通常表示连接到入口接口的链路上的物理层问题）还是堆积CRC（表示连接到入口接口的设备也在直通交换模式下运行并且收到了格式错误的以太网帧）的步骤。

适用硬件

本文档中介绍的过程仅适用于以下硬件：

- **Nexus 9200/9300固定交换机** N9K-C92160YC-XN9K-C92300YCN9K-C92304QCN9K-C92348GC-XN9K-C9236CN9K-C9272QN9K-C9332CN9K-C9364CN9K-C93108TC-EXN9K-C93108TC-EX-24N9K-C93180LC-EXN9K-C93180YC-EXN9K-C93180YC-EX-24N9K-C93108TC-FXN9K-C93108TC-FX-24N9K-C93180YC-FXN9K-C93180YC-FX-24N9K-C9348GC-FXPN9K-C93240YC-FX2N9K-C93216TC-FX2N9K-C9336C-FX2N9K-C9336C-FX2-EN9K-C93360YC-FX2N9K-C93180YC-FX3N9K-C93108TC-FX3PN9K-C93180YC-FX3SN9K-C9316D-GXN9K-C93600CD-GXN9K-C9364C-GXN9K-C9364D-GX2AN9K-C9332D-GX2B
- **Nexus 9500模块化交换机线卡** N9K-X97160YC-EXN9K-X9732C-EXN9K-X9736C-EXN9K-X97284YC-FXN9K-X9732C-FXN9K-X9788TC-FXN9K-X9716D-GX

Cisco Nexus 9200和9300云规模CRC识别和跟踪程序

本文档的此部分将介绍用于识别Cisco Nexus 9200和9300系列交换机上特定物理接口Ethernet1/1上观察到的CRC错误来源的分步说明。

NX-OS软件版本10.2(1)及更高版本

从NX-OS软件版本10.2(1)开始，配备云扩展ASIC的Nexus交换机在流经交换机的以太网帧的FCS字段中为具有存储的CRC的数据包提供新的接口计数器。您可以使用**show interface**命令来识别具有递增非零CRC和堆积CRC计数器的物理接口。这里显示了一个示例，其中物理接口Ethernet1/1具有零CRC计数器和非零堆积CRC计数器，这表示在此接口上接收了具有无效和堆积CRC的帧。

```
switch# show interface
<snip>
Ethernet1/1 is up
admin state is up, Dedicated Interface
Hardware: 100/1000/10000/25000 Ethernet, address: 00d7.8f86.2bbe (bia 00d7.8f86.2bbe)
MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, medium is broadcast
Port mode is trunk
full-duplex, 10 Gb/s, media type is 10G
Beacon is turned off
Auto-Negotiation is turned on FEC mode is Auto
Input flow-control is off, output flow-control is off
Auto-mdix is turned off
Rate mode is dedicated
Switchport monitor is off
EtherType is 0x8100
EEE (efficient-ethernet) : n/a
admin fec state is auto, oper fec state is off
Last link flapped 04:09:21
Last clearing of "show interface" counters 00:50:37
0 interface resets
 RX
   8 unicast packets  253 multicast packets  2 broadcast packets
   1832838280 input packets  2199405650587 bytes
   0 jumbo packets  0 storm suppression bytes
   0 runts  0 giants 1832838019 CRC  0 no buffer
   1832838019 input error  0 short frame  0 overrun   0 underrun  0 ignored
   0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
```

```
   0 input with dribble  0 input discard
   0 Rx pause
   1832838019 Stomped CRC
 TX
   908 unicast packets  323 multicast packets  3 broadcast packets
   1234 output packets  113342 bytes
   0 jumbo packets
   0 output error  0 collision  0 deferred  0 late collision
   0 lost carrier  0 no carrier  0 babble  0 output discard
   0 Tx pause
```

请注意，递增的“CRC”计数器表示收到的帧带有存储的CRC或无效但非存储的CRC。递增的“stopted CRC”计数器增加表示收到具有堆叠CRC的帧。

或者，**show interface counters errors non-zero**命令可用于查看接口错误计数器。这里显示了一个示例。

```
switch# show interface counters errors non-zero
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1      1790348828  1790348828         0  1790348828         0           0

--------------------------------------------------------------------------------
Port         Single-Col  Multi-Col   Late-Col  Exces-Col  Carri-Sen       Runts
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port          Giants SQETest-Err Deferred-Tx IntMacTx-Er IntMacRx-Er Symbol-Err
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port         InDiscards
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port         Stomped-CRC
--------------------------------------------------------------------------------
Eth1/1      1790348828
```

您可以将**show interface**命令管道传输到**json**或**json-pretty**命令，以结构化格式获取CRC和压缩的CRC计数器统计信息。这里显示了一个示例。

```
switch# show interface Ethernet1/1 | json-pretty | include ignore-case crc
           "eth_crc": "828640831",
           "eth_stomped_crc": "828640831",
```

NX-API REST API可用于使用**sys/intf/phys-[intf-id]/dbgEtherStats.json对象模型检索这些相同的统**计信息。这里显示了一个示例。

```
/api/node/mo/sys/intf/phys-[eth1/1]/dbgEtherStats.json
{
 "totalCount": "1",
 "imdata": [
   {
     "rmonEtherStats": {
       "attributes": {
         "cRCAlignErrors": "26874272810",
         "dn": "sys/intf/phys-[eth1/1]/dbgEtherStats",
         "dropEvents": "0",
```

```
        "rXNoErrors": "26874276337",
         "stompedCRCAlignErrors": "26874272810",
    ...
       }
     }
   }
 ]
}
```

NX-OS软件版本10.1(2)及更低版本

对于10.2(1)之前的NX-OS软件版本，堆叠的CRC计数器在接口上不可用。需要执行几个步骤来确定观察到无效CRC的入口接口，并验证CRC是无效还是堆叠。

步骤1.确定物理接口上的CRC计数器递增

使用**show interface**命令识别具有递增的非零CRC计数器的物理接口。此处显示的示例，其中物理接口Ethernet1/1具有非零CRC计数器。

```
switch# show interface
<snip> Ethernet1/1 is up admin state is up, Dedicated Interface Hardware: 100/1000/10000/25000
Ethernet, address: 00d7.8f86.2bbe (bia 00d7.8f86.2bbe) MTU 1500 bytes, BW 10000000 Kbit, DLY 10
usec reliability 255/255, txload 1/255, rxload 1/255 Encapsulation ARPA, medium is broadcast
Port mode is trunk full-duplex, 10 Gb/s, media type is 10G Beacon is turned off Auto-Negotiation
is turned on FEC mode is Auto Input flow-control is off, output flow-control is off Auto-mdix is
turned off Rate mode is dedicated Switchport monitor is off EtherType is 0x8100 EEE (efficient-
ethernet) : n/a admin fec state is auto, oper fec state is off Last link flapped 04:09:21 Last
clearing of "show interface" counters 00:50:37 0 interface resets RX 3 unicast packets 3087
multicast packets 0 broadcast packets 3097 input packets 244636 bytes 7 jumbo packets 0 storm
suppression bytes 0 runts 7 giants 7 CRC  0 no buffer
    7 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
```

或者，您可以使用**show interface counters errors non-zero**命令显示所有带有非零错误计数器（包括非零CRC计数器）的接口。 此处显示的示例，其中物理接口Ethernet1/1的CRC计数器由FCS-Err列显示。

```
switch# show interface counters errors non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                7          7          0          7          0           0
```

步骤2.将物理接口映射到ASIC、MAC块和Mac块子端口

使用**show interface hardware-mappings**命令确定三个主要特征：

1. **Unit** — 物理接口连接的云扩展ASIC的标识符。这使用基于零的编号系统（例如，第一个ASIC是0，第二个ASIC是1等）
2. **MacId** — 物理接口所连接的MAC块的标识符。这使用基于零的编号系统（例如，第一个MAC块是0，第二个MAC块是1等）

3. **MacSP** — 物理接口所连接的MAC块子端口的标识符。每个MAC块都有四个与其关联的子端口，这些子端口遵循基于零的编号系统，并且以2的值递增。因此，第一个子端口的索引为0，第二个子端口的索引为2，第三个子端口的索引为4，第四个子端口的索引为6。

此处的示例说明了这一点，其中物理接口Ethernet1/1与Cloud Scale ASIC 0、MAC块4和MAC块子端口0关联。

```
switch# show interface hardware-mappings
<snip>
-------------------------------------------------------------------------------------------------
-------
Name       Ifindex  Smod Unit HPort FPort NPort VPort Slice SPort SrcId MacId MacSP VIF  Block
BlkSrcID
-------------------------------------------------------------------------------------------------
-------
Eth1/1     1a000000 1    0    16    255   0     -1    0     16    32    4     0     1    0
32
Eth1/2     1a000200 1    0    17    255   4     -1    0     17    34    4     2     5    0
34
Eth1/3     1a000400 1    0    18    255   8     -1    0     18    36    4     4     9    0
36
Eth1/4     1a000600 1    0    19    255   12    -1    0     19    38    4     6     13   0
38
Eth1/5     1a000800 1    0    12    255   16    -1    0     12    24    3     0     17   0
24
```

步骤3.检查CRC相关计数器的云扩展ASIC注册

使用**slot {x} show hardware internal tah counters asic {y}**命令查看云扩展ASIC的注册计数器。此命令包含两个变量：

1. **{x}** — 将此值替换为线卡插槽编号。对于架顶式交换机，值始终为1。对于行尾式模块化交换机，线卡插槽编号将是物理接口名称中的第一个数字。例如，物理接口Ethernet1/1的线卡插槽编号为1，而物理接口Ethernet4/24的线卡插槽编号为4。
2. **{y}** — 用步骤2中标识的云扩展ASIC标识符替换此值。例如，如果物理接口Ethernet1/1的“Unit”列的值为0，则此变量的值为0。如果物理接口Ethernet4/24的“Unit”列的值为3，则此变量的值为3。

此输出将显示一个表。表的每一行都是不同的ASIC寄存器。表中的每一列对应于交换机上的一个物理接口。每列使用的名称不是物理接口的名称，而是MAC块和MAC块子端口的组合。列标题使用的格式如下：

```
M{A},{B}-{InterfaceSpeed}
```

此格式有三个变量，如下所示：

1. **{A}** — 用MAC块编号替换此值。
2. **{B}** — 将此值替换为MAC块子端口号。
3. **{InterfaceSpeed}** — 此值将与接口的物理速度相对应（例如10G、25G、40Gx4等）

此处的示例说明了这一点。回想一下，物理接口Ethernet1/1与线路卡插槽编号1和Cloud Scale ASIC 0关联，这意味着我们必须运行的命令是**slot 1 show hardware internal tah counters asic 0**。与物理接口Ethernet1/1关联的MAC块是4，与物理接口Ethernet1/1关联的MAC块子端口是0，而物理接口Ethernet1/1是10G接口。因此，我们要查找的列标题将是**M4,0-10G**。

注意：以下命令的输出非常长且宽。在终端会话中可能难以读取此输出。Cisco建议使用

terminal width 511命令最大化终端的宽度，并将此输出复制到外部文本阅读器/编辑器以供审阅。

```
switch# slot 1 show hardware internal tah counters asic 0
<snip>
*************** PER MAC/CH SRAM COUNTERS ***************
REG_NAME                  M4,0-10G      M4,2-10G      M4,4-10G      M4,6-10G      M5,0-40Gx4    M6,0-
40Gx4    M7,0-40Gx4    M8,0-10G
-------------------------------------------------------------------------------------------------
----------------------------
02-RX Frm with FCS Err    ....          ....          ....          ....          ....          ....
....          ....
16-RX Frm CRC Err(Stomp) c              ....          ....          ....          ....          ....
....          ....
```

此命令的输出将包含几十个寄存器计数器。有两个关键寄存器计数器与区分自然CRC错误与存储的CRC相关：

1. 02-RX Frm with FCS Err — 表示收到具有无效但未存储的CRC的帧。
2. 16-RX From CRC Err(Stomp)-表示收到具有堆叠CRC的帧。

这些计数器的值为十六进制。dec NX-OS命令可以将十六进制值转换为十进制值，如下所示。

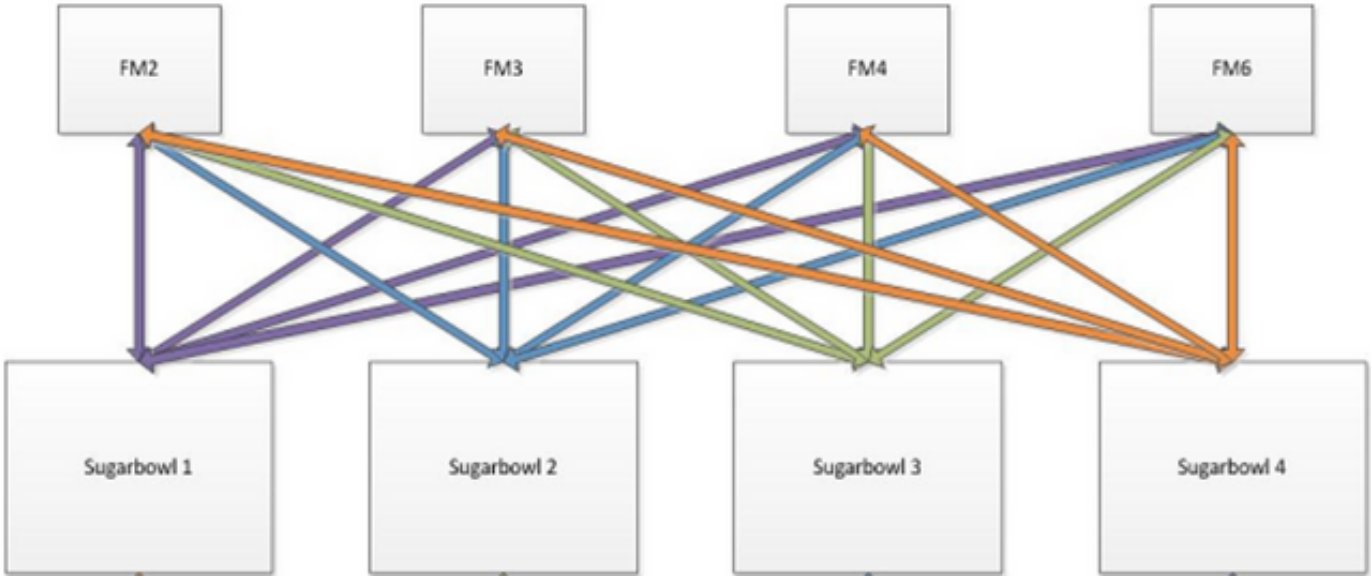
```
N9K-C93180YC-EX-2# dec 0xc
12
```

两个寄存器计数器的组合值将等同于通过show interface或show interface counters errors non-zero的输出在物理接口上观察到的CRC数量。

Cisco Nexus 9500云扩展 — 模块化交换机上的CRC识别和跟踪程序

本文档的此部分将介绍用于识别Cisco Nexus 9500系列交换机上特定物理接口Ethernet1/1上观察到的CRC错误来源的分步说明。

Nexus 9500系列交换机上的每个线卡都通过内部链路（以太网）连接到交换矩阵模块。每个线卡的每个ASIC都与所有交换矩阵模块具有全网状连接。此处的示例显示一个线卡，该线卡带有四个具有内部链路的Sugarbowl ASIC，连接到模块化Nexus 9500交换机中的四个交换矩阵模块。



当ASIC接收的流量需要出口另一个ASIC或线卡时，该流量需要通过交换矩阵模块发送到交换矩阵。入口ASIC将根据数据包报头的散列值以及可用于ASIC的iEth链路数选择到交换矩阵模块的一个iEth链路。

步骤1.映射线卡和交换矩阵模块之间的内部链路。

使用**show system internal fabric connectivity module {x}**命令(其中**{x}是线卡或交换矩阵模块插槽编号)显示指定线卡和所有交换矩阵模块之间的内部链路。**此输出将显示一个表，其中每行显示线卡内部链路（在“LC-iEthLink”列下）与每个交换矩阵模块内部链路（在“FM-iEthLink”列下）之间的一对一映射。 此处显示的是一个示例，取自插入了8个线卡和4个交换矩阵模块的Nexus 9508交换机。此处的输出显示，插入交换机插槽8中的线卡的每个ASIC实例通过2个内部链路连接到4个安装的交换矩阵模块（插入插槽22、23、24和26）中的每个。

```
Nexus9500# show system internal fabric connectivity module 8
Internal Link-info Linecard slot:8
------------------------------------------------------------------------
 LC-Slot  LC-Unit   LC-iEthLink   MUX   FM-Slot   FM-Unit   FM-iEthLink
------------------------------------------------------------------------
   8        0     iEth01         -         22        0      iEth18
   8        0     iEth02         -         22        1      iEth50
   8        0     iEth03         -         23        0      iEth18
   8        0     iEth04         -         23        1      iEth50
   8        0     iEth05         -         24        0      iEth18
   8        0     iEth06         -         24        1      iEth50
   8        0     iEth07         -         26        0      iEth18
   8        0     iEth08         -         26        1      iEth50
   8        1     iEth09         -         22        0      iEth03
   8        1     iEth10         -         22        1      iEth35
   8        1     iEth11         -         23        0      iEth03
   8        1     iEth12         -         23        1      iEth35
   8        1     iEth13         -         24        0      iEth03
   8        1     iEth14         -         24        1      iEth35
   8        1     iEth15         -         26        0      iEth03
   8        1     iEth16         -         26        1      iEth35
   8        2     iEth17         -         22        0      iEth32
   8        2     iEth18         -         22        1      iEth53
   8        2     iEth19         -         23        0      iEth32
   8        2     iEth20         -         23        1      iEth53
   8        2     iEth21         -         24        0      iEth32
   8        2     iEth22         -         24        1      iEth53
```

```
  8        2     iEth23         -         26        0      iEth32
  8        2     iEth24         -         26        1      iEth53
  8        3     iEth25         -         22        0      iEth31
  8        3     iEth26         -         22        1      iEth54
  8        3     iEth27         -         23        0      iEth31
  8        3     iEth28         -         23        1      iEth54
  8        3     iEth29         -         24        0      iEth31
  8        3     iEth30         -         24        1      iEth54
  8        3     iEth31         -         26        0      iEth31
  8        3     iEth32         -         26        1      iEth54
```

同样，可以从交换矩阵模块的角度检查以太网链路映射。这里显示了一个示例，其中显示了插入插槽22中的交换矩阵模块与安装在Nexus 9508机箱中的8个线卡中的每一个之间的内部链路。

```
Nexus9500# show system internal fabric connectivity module 22
Internal Link-info Fabriccard slot:22
------------------------------------------------------------------------
 FM-Slot  FM-Unit   FM-iEthLink   LC-Slot   LC-Unit   LC-EthLink    MUX
------------------------------------------------------------------------
  22         0      iEth09          1         0      iEth01         -
  22         0      iEth06          1         1      iEth11         -
  22         0      iEth25          1         2      iEth21         -
  22         0      iEth26          1         3      iEth31         -
  22         0      iEth10          2         0      iEth01         -
  22         0      iEth05          2         1      iEth11         -
  22         0      iEth23          2         2      iEth21         -
  22         0      iEth24          2         3      iEth31         -
  22         0      iEth12          3         0      iEth01         -
  22         0      iEth11          3         1      iEth11         -
  22         0      iEth21          3         2      iEth21         -
  22         0      iEth22          3         3      iEth31         -
  22         0      iEth14          4         0      iEth01         -
  22         0      iEth13          4         1      iEth11         -
  22         0      iEth07          4         2      iEth21         -
  22         0      iEth08          4         3      iEth31         -
  22         0      iEth16          5         0      iEth01         -
  22         0      iEth15          5         1      iEth11         -
  22         0      iEth01          5         2      iEth21         -
  22         0      iEth04          5         3      iEth31         -
  22         0      iEth20          6         0      iEth01         -
  22         0      iEth17          6         1      iEth11         -
  22         0      iEth28          6         2      iEth21         -
  22         0      iEth27          6         3      iEth31         -
  22         0      iEth19          7         0      iEth01         -
  22         0      iEth02          7         1      iEth09         -
  22         0      iEth30          7         2      iEth17         -
  22         0      iEth29          7         3      iEth25         -
  22         0      iEth18          8         0      iEth01         -
  22         0      iEth03          8         1      iEth09         -
  22         0      iEth32          8         2      iEth17         -
  22         0      iEth31          8         3      iEth25         -
  22         1      iEth41          1         0      iEth02         -
  22         1      iEth38          1         1      iEth12         -
  22         1      iEth59          1         2      iEth22         -
  22         1      iEth60          1         3      iEth32         -
  22         1      iEth42          2         0      iEth02         -
  22         1      iEth37          2         1      iEth12         -
  22         1      iEth62          2         2      iEth22         -
  22         1      iEth61          2         3      iEth32         -
  22         1      iEth44          3         0      iEth02         -
  22         1      iEth43          3         1      iEth12         -
  22         1      iEth64          3         2      iEth22         -
  22         1      iEth63          3         3      iEth32         -
```

```
  22         1      iEth46          4         0      iEth02         -
  22         1      iEth45          4         1      iEth12         -
  22         1      iEth39          4         2      iEth22         -
  22         1      iEth40          4         3      iEth32         -
  22         1      iEth48          5         0      iEth02         -
  22         1      iEth47          5         1      iEth12         -
  22         1      iEth36          5         2      iEth22         -
  22         1      iEth33          5         3      iEth32         -
  22         1      iEth52          6         0      iEth02         -
  22         1      iEth49          6         1      iEth12         -
  22         1      iEth57          6         2      iEth22         -
  22         1      iEth58          6         3      iEth32         -
  22         1      iEth34          7         0      iEth02         -
  22         1      iEth51          7         1      iEth10         -
  22         1      iEth55          7         2      iEth18         -
  22         1      iEth56          7         3      iEth26         -
  22         1      iEth50          8         0      iEth02         -
  22         1      iEth35          8         1      iEth10         -
  22         1      iEth53          8         2      iEth18         -
  22         1      iEth54          8         3      iEth26         -
```

使用**show system internal fabric link-state module {x}**命令检查内部端口是否为up（在“ST”列下），以及特定内部链路（在“MAC”列下）的对应ASIC片和MAC标识符是什么。这里显示了一个示例。

```
Nexus9500# show system internal fabric link-state module 8
 cli : mod = 8
 module number = 8
================================================================================================================
 Module number = 8
================================================================================================================
 [LC]   [    INST:SLI:MAC:GLSRC]   [IETH]    [ST]   <========>  [FM]   [    INST:SLI:MAC:GLSRC]   [IETH]   [ST]
================================================================================================================
 [ 8]   [  0 :   0 :   7 : 0x38]   [iEth01]   [UP]  <========>  [22]   [  0 :   3 :  21 : 0x18]   [iEth18]   [UP]
 [ 8]   [  0 :   1 :   9 :  0x0]   [iEth02]   [UP]  <========>  [22]   [  1 :   3 :  21 : 0x18]   [iEth50]   [UP]
 [ 8]   [  0 :   0 :   6 : 0x30]   [iEth03]   [UP]  <========>  [23]   [  0 :   3 :  21 : 0x18]   [iEth18]   [UP]
 [ 8]   [  0 :   1 :  16 : 0x38]   [iEth04]   [UP]  <========>  [23]   [  1 :   3 :  21 : 0x18]   [iEth50]   [UP]
 [ 8]   [  0 :   0 :   8 : 0x40]   [iEth05]   [UP]  <========>  [24]   [  0 :   3 :  21 : 0x18]   [iEth18]   [UP]
 [ 8]   [  0 :   1 :  15 : 0x30]   [iEth06]   [UP]  <========>  [24]   [  1 :   3 :  21 : 0x18]   [iEth50]   [UP]
 [ 8]   [  0 :   0 :   5 : 0x28]   [iEth07]   [UP]  <========>  [26]   [  0 :   3 :  21 : 0x18]   [iEth18]   [UP]
 [ 8]   [  0 :   1 :  17 : 0x40]   [iEth08]   [UP]  <========>  [26]   [  1 :   3 :  21 : 0x18]   [iEth50]   [UP]
 [ 8]   [  1 :   0 :   7 : 0x38]   [iEth09]   [UP]  <========>  [22]   [  0 :   0 :   4 : 0x20]   [iEth03]   [UP]
 [ 8]   [  1 :   1 :   9 :  0x0]   [iEth10]   [UP]  <========>  [22]   [  1 :   0 :   4 : 0x20]   [iEth35]   [UP]
 [ 8]   [  1 :   0 :   6 : 0x30]   [iEth11]   [UP]  <========>  [23]   [  0 :   0 :   4 : 0x20]   [iEth03]   [UP]
 [ 8]   [  1 :   1 :  16 : 0x38]   [iEth12]   [UP]  <========>  [23]   [  1 :   0 :   4 : 0x20]   [iEth35]   [UP]
 [ 8]   [  1 :   0 :   8 : 0x40]   [iEth13]   [UP]  <========>  [24]   [  0 :   0 :   4 : 0x20]   [iEth03]   [UP]
```

```
 [ 8]   [  1 :   1 :  15 : 0x30]   [iEth14]   [UP]  <========>  [24]   [  1 :   0 :   4 :
0x20]   [iEth35]   [UP]
 [ 8]   [  1 :   0 :   5 : 0x28]   [iEth15]   [UP]  <========>  [26]   [  0 :   0 :   4 :
0x20]   [iEth03]   [UP]
 [ 8]   [  1 :   1 :  17 : 0x40]   [iEth16]   [UP]  <========>  [26]   [  1 :   0 :   4 :
0x20]   [iEth35]   [UP]
 [ 8]   [  2 :   0 :   7 : 0x38]   [iEth17]   [UP]  <========>  [22]   [  0 :   5 :  35 :
0x28]   [iEth32]   [UP]
 [ 8]   [  2 :   1 :   9 :  0x0]   [iEth18]   [UP]  <========>  [22]   [  1 :   4 :  24 :
0x0]   [iEth53]   [UP]
 [ 8]   [  2 :   0 :   6 : 0x30]   [iEth19]   [UP]  <========>  [23]   [  0 :   5 :  35 :
0x28]   [iEth32]   [UP]
 [ 8]   [  2 :   1 :  16 : 0x38]   [iEth20]   [UP]  <========>  [23]   [  1 :   4 :  24 :
0x0]   [iEth53]   [UP]
 [ 8]   [  2 :   0 :   8 : 0x40]   [iEth21]   [UP]  <========>  [24]   [  0 :   5 :  35 :
0x28]   [iEth32]   [UP]
 [ 8]   [  2 :   1 :  15 : 0x30]   [iEth22]   [UP]  <========>  [24]   [  1 :   4 :  24 :
0x0]   [iEth53]   [UP]
 [ 8]   [  2 :   0 :   5 : 0x28]   [iEth23]   [UP]  <========>  [26]   [  0 :   5 :  35 :
0x28]   [iEth32]   [UP]
 [ 8]   [  2 :   1 :  17 : 0x40]   [iEth24]   [UP]  <========>  [26]   [  1 :   4 :  24 :
0x0]   [iEth53]   [UP]
 [ 8]   [  3 :   0 :   7 : 0x38]   [iEth25]   [UP]  <========>  [22]   [  0 :   5 :  34 :
0x20]   [iEth31]   [UP]
 [ 8]   [  3 :   1 :   9 :  0x0]   [iEth26]   [UP]  <========>  [22]   [  1 :   4 :  25 :
0x8]   [iEth54]   [UP]
 [ 8]   [  3 :   0 :   6 : 0x30]   [iEth27]   [UP]  <========>  [23]   [  0 :   5 :  34 :
0x20]   [iEth31]   [UP]
 [ 8]   [  3 :   1 :  16 : 0x38]   [iEth28]   [UP]  <========>  [23]   [  1 :   4 :  25 :
0x8]   [iEth54]   [UP]
 [ 8]   [  3 :   0 :   8 : 0x40]   [iEth29]   [UP]  <========>  [24]   [  0 :   5 :  34 :
0x20]   [iEth31]   [UP]
 [ 8]   [  3 :   1 :  15 : 0x30]   [iEth30]   [UP]  <========>  [24]   [  1 :   4 :  25 :
0x8]   [iEth54]   [UP]
 [ 8]   [  3 :   0 :   5 : 0x28]   [iEth31]   [UP]  <========>  [26]   [  0 :   5 :  34 :
0x20]   [iEth31]   [UP]
 [ 8]   [  3 :   1 :  17 : 0x40]   [iEth32]   [UP]  <========>  [26]   [  1 :   4 :  25 :
0x8]   [iEth54]   [UP]
```

步骤2.检查iEth链路上的CRC计数器并跟踪损坏帧的来源。

在模块化Nexus 9500交换机上，您可能会在以下场景中看到一个或多个iEth链路上出现CRC错误：

1. 当交换机以直通交换模式运行时，在FCS字段中收到损坏的以太网帧且CRC值不正确的线卡不会本地丢弃线卡。线路卡将正常转发数据包。如果数据包的出口接口属于另一个ASIC或线卡，入口线卡会将数据包转发到交换矩阵模块。交换矩阵模块也在直通交换模式下运行，因此交换矩阵模块会将数据包转发到出口线卡。出口线卡会将数据包转发到下一跳，并增加出口接口上的输出错误计数器。
2. 如果内部链路由于硬件故障而失败，则流经内部链路的数据包可能在线卡和交换矩阵模块之间损坏。

使用**show system internal fabric connectivity stats module {x}**命令检查对应内部链路的CRC计数器。此处显示了一个示例，插入插槽22中的交换矩阵模块接收与插入交换机插槽7中的线路卡的iEth26连接的iEth56上具有无效CRC的数据包。这表示交换矩阵模块从插入交换机插槽7的线卡接收损坏的以太网帧。

```
Nexus9500# show system internal fabric connectivity stats module 22
Internal Link-info Stats Fabriccard slot:22
--------------------------------------------------------------------------------
```

FM-Slot	FM-Unit	FM-iEthLink	LC-Slot	LC-Unit	LC-EthLink	MUX	CRC
22	0	iEth09	1	0	iEth01	-	0
22	0	iEth06	1	1	iEth11	-	0
22	0	iEth25	1	2	iEth21	-	0
22	0	iEth26	1	3	iEth31	-	0
22	0	iEth10	2	0	iEth01	-	0
22	0	iEth05	2	1	iEth11	-	0
22	0	iEth23	2	2	iEth21	-	0
22	0	iEth24	2	3	iEth31	-	0
22	0	iEth12	3	0	iEth01	-	0
22	0	iEth11	3	1	iEth11	-	0
22	0	iEth21	3	2	iEth21	-	0
22	0	iEth22	3	3	iEth31	-	0
22	0	iEth14	4	0	iEth01	-	0
22	0	iEth13	4	1	iEth11	-	0
22	0	iEth07	4	2	iEth21	-	0
22	0	iEth08	4	3	iEth31	-	0
22	0	iEth16	5	0	iEth01	-	0
22	0	iEth15	5	1	iEth11	-	0
22	0	iEth01	5	2	iEth21	-	0
22	0	iEth04	5	3	iEth31	-	0
22	0	iEth20	6	0	iEth01	-	0
22	0	iEth17	6	1	iEth11	-	0
22	0	iEth28	6	2	iEth21	-	0
22	0	iEth27	6	3	iEth31	-	0
22	0	iEth19	7	0	iEth01	-	0
22	0	iEth02	7	1	iEth09	-	0
22	0	iEth30	7	2	iEth17	-	0
22	0	iEth29	7	3	iEth25	-	0
22	0	iEth18	8	0	iEth01	-	0
22	0	iEth03	8	1	iEth09	-	0
22	0	iEth32	8	2	iEth17	-	0
22	0	iEth31	8	3	iEth25	-	0
22	1	iEth41	1	0	iEth02	-	0
22	1	iEth38	1	1	iEth12	-	0
22	1	iEth59	1	2	iEth22	-	0
22	1	iEth60	1	3	iEth32	-	0
22	1	iEth42	2	0	iEth02	-	0
22	1	iEth37	2	1	iEth12	-	0
22	1	iEth62	2	2	iEth22	-	0
22	1	iEth61	2	3	iEth32	-	0
22	1	iEth44	3	0	iEth02	-	0
22	1	iEth43	3	1	iEth12	-	0
22	1	iEth64	3	2	iEth22	-	0
22	1	iEth63	3	3	iEth32	-	0
22	1	iEth46	4	0	iEth02	-	0
22	1	iEth45	4	1	iEth12	-	0
22	1	iEth39	4	2	iEth22	-	0
22	1	iEth40	4	3	iEth32	-	0
22	1	iEth48	5	0	iEth02	-	0
22	1	iEth47	5	1	iEth12	-	0
22	1	iEth36	5	2	iEth22	-	0
22	1	iEth33	5	3	iEth32	-	0
22	1	iEth52	6	0	iEth02	-	0
22	1	iEth49	6	1	iEth12	-	0
22	1	iEth57	6	2	iEth22	-	0
22	1	iEth58	6	3	iEth32	-	0
22	1	iEth34	7	0	iEth02	-	0
22	1	iEth51	7	1	iEth10	-	0
22	1	iEth55	7	2	iEth18	-	0
22	**1**	**iEth56**	**7**	**3**	**iEth26**	-	**1665601166**
22	1	iEth50	8	0	iEth02	-	0
22	1	iEth35	8	1	iEth10	-	0

```
  22         1      iEth53          8         2      iEth18          -     0
  22         1      iEth54          8         3      iEth26          -     0
```

在板卡或交换矩阵模块上使用**slot {x} show hardware internal tah counters asic {y}**命令确定CRC错误是无效还是存储的CRC。将无效CRC错误与存储的CRC错误区分开来的两个寄存器计数器如下：

1. **02-RX Frm with FCS Err** — 表示收到具有无效但未存储的CRC的帧。
2. **16-RX From CRC Err(Stomp)-**表示收到具有堆叠CRC的帧。

这里显示了一个示例，其中通过内部链路iEth54在插入机箱插槽22的交换矩阵模块上接收的损坏帧通过堆叠的CRC与插入机箱插槽8的线卡连接：

```
Nexus9500# slot 22 show hardware internal tah counters asic 1
REG_NAME                  M24,0-
100Gx4                                                                                  M25,0-100Gx4
-------------------------------------------------------------------------------------------------------
-------------------------
02-RX Frm with FCS Err
....                                                                                          ....
03-RX Frm with any Err
....                                                                                          144912733e
16-RX Frm CRC Err(Stomp)
....                                                                                          14491277d7
```

或者，使用**show hardware internal errors module {x}**命令查看特定模块的ASIC错误计数器。这里显示了一个示例。请注意，在此输出中，“接口入站错误(CRC、len、Algn Err)”计数器为无效CRC和堆积CRC增加，而“接口入站CRC错误堆积”计数器仅为堆积CRC增加。

```
Nexus9500# show hardware internal errors module 22
|-------------------------------------------------------------------------|
| Device:Lacrosse                 Role:MAC                      Mod:22    |
| Last cleared @ Tue Jul  6 04:10:45 2021
| Device Statistics Category :: ERROR
|-------------------------------------------------------------------------|
Instance:0
ID    Name                                          Value             Ports
--    ----                                          -----             -----

Instance:1
ID    Name                                          Value             Ports
--    ----                                          -----             -----
196635  Interface Inbound Errors (CRC,len,Algn Err)   0000053053264536  27:0
1048603 Interface Inbound CRC Error Stomped           0000053053264535  27:0
```

在识别接收损坏帧的入口线卡后，以类似的方式使用**slot {x} show hardware internal tah counters asic {y}**或**show hardware internal errors module {x}**命令来识别接收错误的入口接口，以及是否将错误作为无效CRC或存储的CRC接收。

交换矩阵模块或出口线卡在iEth链路上显示CRC错误，但连接的线卡没有入口CRC符号的情况极少。此问题的根源通常是交换矩阵模块的硬件故障。思科建议使用Cisco TAC创建支持案例以进一步解决此问题，并在必要时更换交换矩阵模块。

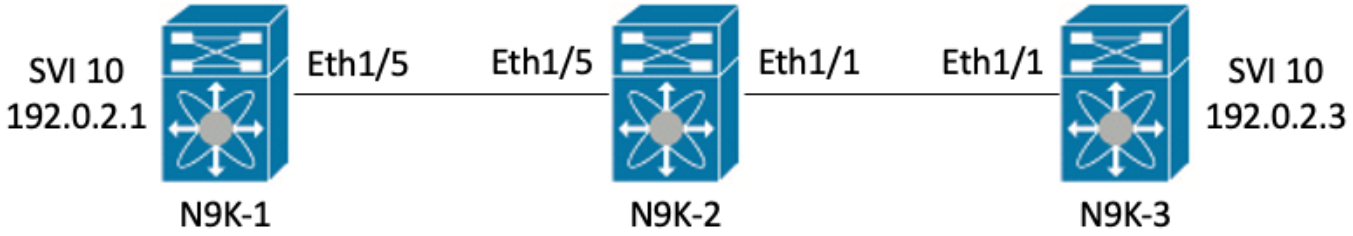
Examples

本文档的此部分将通过一些示例完成上述步骤。

场景1.接收堆叠式CRC的物理接口

此示例演示如何识别物理接口上的CRC错误是存储的CRC。

请考虑以下拓扑：



在本示例中，通过从接口SVI 10（拥有IP地址192.0.2.1）发往N9K-3的接口SVI 10（拥有IP地址192.0.2.3）（拥有MTU 1500字节）的巨型8000字节ICMP数据包，在交换机N9K-1上生成有意存储的CRC错误。N9K-1、N9K-2和N9K-3都是Nexus 93180YC-EX型号交换机。

```
N9K-3# ping 192.0.2.3 count 5 packet-size 8000
PING 192.0.2.3 (192.0.2.3): 8000 data bytes
Request 0 timed out
Request 1 timed out
Request 2 timed out
Request 3 timed out
Request 4 timed out
Request 5 timed out

--- 192.0.2.3 ping statistics ---
5 packets transmitted, 0 packets received, 100.00% packet loss
```

在本例中，观察到交换机N9K-3的物理接口Ethernet1/1上出现递增的CRC错误。

```
N9K-3# show interface Ethernet1/1
<snip>
Ethernet1/1 is up
admin state is up, Dedicated Interface
  Hardware: 100/1000/10000/25000 Ethernet, address: 00d7.8f86.2bbe (bia 00d7.8f86.2bbe)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 10G
  Beacon is turned off
  Auto-Negotiation is turned on  FEC mode is Auto
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
    admin fec state is auto, oper fec state is off
  Last link flapped 06:13:44
  Last clearing of "show interface" counters 02:55:00
  0 interface resets
  RX
    9 unicast packets  10675 multicast packets  0 broadcast packets
    10691 input packets  816924 bytes
    7 jumbo packets  0 storm suppression bytes
    0 runts  7 giants  7 CRC  0 no buffer
```

```
    7 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
```

步骤1.确认增加CRC

通过生成从N9K-1的接口SVI 10（拥有IP地址192.0.2.1）到N9K-3的接口SVI 10（拥有IP地址192.0.2.3）的巨型8000字节ICMP数据包，确认CRC在物理接口Ethernet1/1上增加。

```
N9K-1# ping 192.0.2.3 count 5 packet-size 8000
PING 192.0.2.3 (192.0.2.3): 8000 data bytes
Request 0 timed out
Request 1 timed out
Request 2 timed out
Request 3 timed out
Request 4 timed out
Request 5 timed out

--- 192.0.2.3 ping statistics ---
5 packets transmitted, 0 packets received, 100.00% packet loss

N9K-3# show interface Ethernet1/1

Ethernet1/1 is up
admin state is up, Dedicated Interface
  Hardware: 100/1000/10000/25000 Ethernet, address: 00d7.8f86.2bbe (bia 00d7.8f86.2bbe)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 10G
  Beacon is turned off
  Auto-Negotiation is turned on  FEC mode is Auto
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
    admin fec state is auto, oper fec state is off
  Last link flapped 06:52:57
  Last clearing of "show interface" counters 03:34:13
  0 interface resets
  RX
    11 unicast packets  13066 multicast packets  0 broadcast packets
    13089 input packets  1005576 bytes
    12 jumbo packets  0 storm suppression bytes
    0 runts  12 giants  12 CRC  0 no buffer
    12 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
```

步骤2.将物理接口映射到ASIC、MAC块和MAC块子端口

在N9K-3上使用**show interface hardware-mappings**命令将物理接口Ethernet1/1映射到ASIC编号0、MAC块4和MAC块子端口0。

```
N9K-3# show interface hardware-mappings
<snip>
-------------------------------------------------------------------------------------------------
-------
Name       Ifindex  Smod Unit HPort FPort NPort VPort Slice SPort SrcId MacId MacSP VIF  Block
BlkSrcID
-------------------------------------------------------------------------------------------------
-------
Eth1/1     1a000000 1    0    16    255   0     -1    0     16    32    4     0     1    0
32
Eth1/2     1a000200 1    0    17    255   4     -1    0     17    34    4     2     5    0
34
Eth1/3     1a000400 1    0    18    255   8     -1    0     18    36    4     4     9    0
36
Eth1/4     1a000600 1    0    19    255   12    -1    0     19    38    4     6     13   0
38
Eth1/5     1a000800 1    0    12    255   16    -1    0     12    24    3     0     17   0
24
```

步骤3.检查CRC相关计数器的云扩展ASIC注册

根据步骤2中的信息，我们知道以下事实：

1. 物理接口Ethernet1/1映射到ASIC编号0。
2. 物理接口Ethernet1/1映射到MAC块4的MAC块子端口0
3. 由于N9K-3是架顶式Nexus 93180YC-EX型号交换机，因此我们知道唯一可能的线卡插槽编号是1
4. 从步骤1中收集的show interface输出中，我们可以知道物理接口Ethernet1/1的速度是10G。

使用此信息，我们可以使用**slot 1 show hardware internal tah counters asic 0**命令查看所有物理接口的ASIC寄存器计数器。具体而言，我们将寻找与M4、0-10G关联的ASIC寄存器计数器。

```
N9K-3# slot 1 show hardware internal tah counters asic 0
<snip>
*************** PER MAC/CH SRAM COUNTERS ***************
REG_NAME                 M4,0-10G     M4,2-10G     M4,4-10G     M4,6-10G     M5,0-40Gx4
M6,0-40Gx4     M7,0-40Gx4     M8,0-10G
-------------------------------------------------------------------------------------------------
---------------------------------
02-RX Frm with FCS Err   ....         ....         ....         ....         ....           ....
....           ....
16-RX Frm CRC Err(Stomp) c            ....         ....         ....         ....           ....
....           ....
```

我们可以看到寄存器16的非零十六进制值0xc，表示此物理接口上收到了具有堆积CRC的帧。我们可以使用**dec 0xc**命令将此值转换为十进制值12，该值与物理接口Ethernet1/1上的CRC错误数匹配。

```
N9K-3# dec 0xc
12
```

场景1总结

我们确认N9K-3在物理接口Ethernet1/1上接收带有堆积CRC的帧。这意味着Ethernet1/1链路远程端的设备（在本例中为N9K-2）正在堆积CRC;格式错误的帧的根源不是直接连接到Ethernet1/1的链路，而是更下游的链路。应对下游网络设备执行更多故障排除，以确定这些错误帧的来源。

场景2.物理接口收到格式错误的帧，其CRC无效

此示例演示如何识别由于直连链路上的物理层问题导致帧格式错误，导致物理接口上的CRC错误增加。

请考虑以下拓扑：

在本示例中，连接到交换机N9K-1的物理接口Ethernet1/40的流量生成器会故意生成具有错误CRC的帧。这将模拟连接到Ethernet1/40的链路上的物理层问题，例如收发器故障或电缆损坏。N9K-1接收这些帧，识别CRC无效，并增加Ethernet1/40物理接口上的CRC错误计数器。N9K-1是Nexus 93180YC-EX型号交换机。

```
N9K-1# show interface Ethernet1/40

Ethernet1/40 is up
admin state is up, Dedicated Interface
  Hardware: 100/1000/10000/25000 Ethernet, address: 00d7.8f86.2bbe (bia 00d7.8f86.2c02)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 10G
  Beacon is turned off
  Auto-Negotiation is turned on  FEC mode is Auto
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
    admin fec state is auto, oper fec state is off
  Last link flapped 06:13:44
  Last clearing of "show interface" counters 02:55:00
  0 interface resets
  RX
    1710 unicast packets  9873 multicast packets  0 broadcast packets
    11583 input packets  886321 bytes
    0 jumbo packets  0 storm suppression bytes
    0 runts  0 giants  1683 CRC  0 no buffer
    1683 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
```

步骤1.确认增加CRC

通过**show interface**或**show interface counters non-zero**命令确认N9K-1的物理接口Ethernet1/40上的CRC正在增加。

```
N9K-1# show interface Ethernet1/40
<snip>
Ethernet1/40 is up
admin state is up, Dedicated Interface
  Hardware: 100/1000/10000/25000 Ethernet, address: 00d7.8f86.2bbe (bia 00d7.8f86.2c02)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 10G
  Beacon is turned off
  Auto-Negotiation is turned on  FEC mode is Auto
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
    admin fec state is auto, oper fec state is off
  Last link flapped 06:13:44
  Last clearing of "show interface" counters 02:55:00
  0 interface resets
  RX
    14055 unicast packets  9873 multicast packets  0 broadcast packets
    23928 input packets  1676401 bytes
    0 jumbo packets  0 storm suppression bytes
    0 runts  0 giants  14028 CRC  0 no buffer
    14028 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause

N9K-1# show interface counters errors non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/40           26373      26373          0      26373          0           0
```

步骤2.将物理接口映射到ASIC、MAC块和MAC块子端口

在N9K-1上使用**show interface hardware-mappings**命令将物理接口Ethernet1/40映射到ASIC编号0、MAC块10、MAC块子端口6。

```
N9K-1# show interface hardware-mappings
<snip>
-----------------------------------------------------------------------------------------------
Name       Ifindex  Smod Unit HPort FPort NPort VPort Slice SPort SrcId MacId MacSP VIF  Block
BlkSrcID
-----------------------------------------------------------------------------------------------
Eth1/38    1a004a00 1    0    45    255   148   -1    1     5     10    10    2     149  0
10
Eth1/39    1a004c00 1    0    46    255   152   -1    1     6     12    10    4     153  0
```

```
12
Eth1/40     1a004e00 1    0     47    255   156   -1    1     7     14    10    6     157   0
14
Eth1/41     1a005000 1    0     76    255   160   -1    1     36    64    17    0     161   0
64
Eth1/42     1a005200 1    0     77    255   164   -1    1     37    66    17    2     165   0
66
```

步骤3.检查CRC相关计数器的云扩展ASIC注册

根据步骤2中的信息，我们知道以下事实：

1. 物理接口Ethernet1/40映射到ASIC编号0。
2. 物理接口Ethernet1/40映射到MAC块10的MAC块子端口6。
3. 由于N9K-1是架顶式Nexus 93180YC-EX型号交换机，因此我们知道唯一可能的线卡插槽编号是1。
4. 从步骤1中收集的**show interface**输出中，我们可以知道物理接口Ethernet1/40的速度是10G。

使用此信息，我们可以使用slot 1 show hardware internal tah counters asic 0命令查看所有物理接口的ASIC寄存器计数器。具体而言，我们将寻找与M10、6-10G关联的ASIC寄存器计数器。

```
N9K-1# slot 1 show hardware internal tah counters asic 0

*************** PER MAC/CH SRAM COUNTERS ****************
REG_NAME                  M8,2-10G     M8,4-10G     M8,6-10G     M9,0-40Gx4     M10,0-10G
M10,2-10G      M10,4-10G      M10,6-10G
-------------------------------------------------------------------------------------------------
-----------------------------------
02-RX Frm with FCS Err   ....          ....          ....          ....           ....
....           ....           973e
16-RX Frm CRC Err(Stomp) ....          ....          ....          ....           ....
....           ....           ....
```

我们可以看到寄存器2的非零十六进制值0x973e，这表示在此物理接口上接收到具有无效但非堆积CRC的帧。我们可以使用dec 0x973e命令将此值转换为38,718的十进制值，该值匹配（或小于，因为CRC不断增大）物理接口Ethernet1/40上的CRC错误数。

```
N9K-1# dec 0x973e
38718
```

场景2总结

我们已经确认，N9K-1正在物理接口Ethernet1/40上接收具有无效但未存储的CRC的帧。这意味着直接连接到Ethernet1/40的链路（或链路远端设备）最可能是格式错误的帧的来源。应该对此链路的物理层执行进一步的故障排除，以查明帧格式错误的根本原因（例如检查电缆是否损坏、使用已知正常的收发器更换当前收发器等）。

场景3. Nexus 9500 iEth CRC错误系统日志

此示例演示当Nexus 9500系列交换机生成内部接口上的系统日志报告错误时，如何识别iEth内部链路上的CRC错误来源。此系统日志的示例如下所示。

```
Nexus9500# show logging logfile
```

```
<snip>
2021 Jul  9 05:51:19 Nexus9500 %DEVICE_TEST-SLOT22-3-INTERNAL_PORT_MONITOR_CRC_ERRORS_DETECTED:
Module 22 received tx errors on internal interface ii22/1/56 since last run TXErr=36836897
TotalTXErr=50781987904
```

此系统日志指示在插入交换机插槽22中的交换矩阵模块的iEth56内部链路上检测到错误。

步骤1.将交换矩阵模块上的以太网链路映射到连接的线卡

使用**show system internal fabric connectivity stats module {x}**命令确定受影响的Eth内部链路连接到的线路卡。在本示例中，插入交换机插槽22中的交换矩阵模块的iEth56出现错误。这里显示了一个示例，其中插入插槽22中的交换矩阵模块的iEth56连接到插入交换机插槽7中的线路卡的iEth26。

```
Nexus9500# show system internal fabric connectivity stats module 22 | include Eth56|FM-Slot
FM-Slot  FM-Unit   FM-iEthLink   LC-Slot   LC-Unit   LC-EthLink    MUX    CRC
  22         1       iEth56          7         3       iEth26          -     603816174
```

使用**show system internal fabric link-state module {x}**命令查找与交换矩阵模块的iEth56内部链路关联的ASIC实例和MAC标识符。这里显示了一个示例，其中ASIC实例为1,MAC标识符为27。

```
Nexus9500# show system internal fabric link-state module 22 | include MAC|iEth56
 [FM]   [    INST:SLI:MAC:GLSRC]   [IETH]    [ST]   <========>  [LC]   [    INST:SLI:MAC:GLSRC]
[IETH]   [ST]
 [22]   [  1 :   4 :  27 : 0x18]   [iEth56]   [UP]  <========>  [ 7]   [  3 :   1 :   9 :
0x0]   [iEth26]   [UP]
```

步骤2.检查以太网链路上收到的CRC是否无效或过大

上一步显示，连接到插入插槽22中的交换矩阵模块的iEth56的ASIC实例标识符为1,MAC标识符为27。使用**slot {x} show hardware internal tah counters asic {y}**命令识别系统日志报告的CRC是无效CRC还是存储的CRC。这里显示了一个示例，其中M27,0-100Gx4列与我们的MAC标识符27关联，表示CRC已堆叠。

```
Nexus9500# slot 22 show hardware internal tah counters asic 1
REG_NAME                  M27,0-100Gx4
----------------------------------------
02-RX Frm with FCS Err    ....
16-RX Frm CRC Err(Stomp)  be9cb9bd6
```

或者，使用**show hardware internal errors module {x}** 命令获取此相同信息。这里显示了一个示例。

```
Nexus9500# show hardware internal errors module 22 | include CRC|Stomp|Inst
Instance:1
196635 Interface Inbound Errors (CRC,len,Algn Err)    0000051587084851  27:0
1048603 Interface Inbound CRC Error Stomped           0000051587084850  27:0
```

回想一下，在此输出中，“接口入站错误(CRC、len、Algn Err)”计数器为无效CRC和堆积CRC增加，而“接口入站CRC错误堆积”计数器仅为堆积CRC增加。

步骤3.跟踪入口线路卡上带有无效CRC的帧的来源

我们现在知道，进入插入交换机插槽22中的交换矩阵模块的CRC是从插入插槽7中的线卡进入交换机。使用此信息，我们可以使用**show interface counters errors module {x} non-zero**命令识别属于相关线卡的接口上的非零CRC计数器。这里显示了一个示例。

```
Nexus9500# show interface counters errors module 7 non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth7/32               0          0          0 1195309745          0           0
```

我们可以在相关线卡上重复此方案的步骤#2，以验证线卡是否接收无效CRC或存储的CRC。

```
Nexus9500# show hardware internal errors module 7 | include ignore-case CRC|Stomp|Inst
Instance:3
196619 Interface Inbound Errors (CRC,len,Algn Err)    0000051801011139  11:0
1048587 Interface Inbound CRC Error Stomped           0000051801011140  11:0
```

使用show interface hardware-mappings命令标识上面输出中的MacId:MacSP值11:0映射到的前面板端口。这里显示一个示例，其中11:0映射到前面板端口Eth7/32。

```
Nexus9500# show interface hardware-mappings | include Name|Eth7
<snip>
Name       Ifindex  Smod Unit HPort FPort NPort VPort Slice SPort SrcId MacId MacSP VIF  Block
BlkSrcID
Eth7/1     1a300000 25   0    16    255   0     -1    0     16    32    4     0     1    0
32
Eth7/2     1a300200 25   0    12    255   4     -1    0     12    24    3     0     5    0
24
Eth7/3     1a300400 25   0    8     255   8     -1    0     8     16    2     0     9    0
16
Eth7/4     1a300600 25   0    4     255   12    -1    0     4     8     1     0     13   0     8
Eth7/5     1a300800 25   0    60    255   16    -1    1     20    40    14    0     17   0
40
Eth7/6     1a300a00 25   0    56    255   20    -1    1     16    32    13    0     21   0
32
Eth7/7     1a300c00 25   0    52    255   24    -1    1     12    24    12    0     25   0
24
Eth7/8     1a300e00 25   0    48    255   28    -1    1     8     16    11    0     29   0
16
Eth7/9     1a301000 26   1    12    255   32    -1    0     12    24    3     0     33   0
24
Eth7/10    1a301200 26   1    8     255   36    -1    0     8     16    2     0     37   0
16
Eth7/11    1a301400 26   1    4     255   40    -1    0     4     8     1     0     41   0     8
Eth7/12    1a301600 26   1    0     255   44    -1    0     0     0     0     0     45   0     0
Eth7/13    1a301800 26   1    60    255   48    -1    1     20    40    14    0     49   0
40
Eth7/14    1a301a00 26   1    56    255   52    -1    1     16    32    13    0     53   0
32
Eth7/15    1a301c00 26   1    52    255   56    -1    1     12    24    12    0     57   0
24
Eth7/16    1a301e00 26   1    48    255   60    -1    1     8     16    11    0     61   0
16
Eth7/17    1a302000 27   2    16    255   64    -1    0     16    32    4     0     65   0
32
Eth7/18    1a302200 27   2    12    255   68    -1    0     12    24    3     0     69   0
24
Eth7/19    1a302400 27   2    8     255   72    -1    0     8     16    2     0     73   0
16
Eth7/20    1a302600 27   2    4     255   76    -1    0     4     8     1     0     77   0     8
Eth7/21    1a302800 27   2    60    255   80    -1    1     20    40    14    0     81   0
40
Eth7/22    1a302a00 27   2    56    255   84    -1    1     16    32    13    0     85   0
32
```

```
Eth7/23    1a302c00 27   2    52    255   88    -1    1     12    24    12    0     89   0
24
Eth7/24    1a302e00 27   2    48    255   92    -1    1     8     16    11    0     93   0
16
Eth7/25    1a303000 28   3    12    255   96    -1    0     12    24    3     0     97   0
24
Eth7/26    1a303200 28   3    8     255   100   -1    0     8     16    2     0     101  0
16
Eth7/27    1a303400 28   3    4     255   104   -1    0     4     8     1     0     105  0     8
Eth7/28    1a303600 28   3    0     255   108   -1    0     0     0     0     0     109  0     0
Eth7/29    1a303800 28   3    60    255   112   -1    1     20    40    14    0     113  0
40
Eth7/30    1a303a00 28   3    56    255   116   -1    1     16    32    13    0     117  0
  32
Eth7/31    1a303c00 28   3    52    255   120   -1    1     12    24    12    0     121  0
24
Eth7/32    1a303e00 28   3    48    255   124   -1    1     8     16    11    0     125  0
16
```

场景3总结

我们已经确认，Nexus 9500在物理接口Ethernet7/32上接收带有堆积CRC的帧。这意味着Ethernet7/32链路远程端的设备正在堆积CRC这些帧；格式错误的帧的根源不是直接连接到Ethernet7/32的链路，而是更下游的链路。应对下游网络设备执行更多故障排除，以确定这些错误帧的来源。

场景4.跟踪具有传出接口的无效CRC帧的来源。

此示例演示当上游交换机报告Nexus 9500正在生成具有堆积CRC的帧时，如何跟踪Nexus 9500交换机上具有无效CRC的帧的源。在此场景中，上游交换机通过前面板端口Ethernet8/9连接。

步骤1.识别将无效CRC帧发送到出口线路卡的交换矩阵模块

我们知道，向上游交换机发送带有堆积CRC的帧的传出接口是Ethernet8/9。首先，我们需要确定将带有堆积CRC的帧发送到插在机箱插槽8中的线卡的交换矩阵模块。我们使用**show hardware internal errors module {x}**命令开始此过程。这里显示了一个示例。

```
Nexus9500# show hardware internal errors module 8 | i CRC|Inst
<snip>
Instance:1
196617 Interface Inbound Errors (CRC,len,Algn Err)    0000091499464650  9:0
1048585 Interface Inbound CRC Error Stomped           0000091499464651  9:0
```

上述输出中的MacID:MacSP 9:0可使用**show system internal fabric link-state module 8命令映射到源交换矩阵模**块。这里显示了一个示例。

```
Nexus9500# show system internal fabric link-state module 8
 cli : mod = 8
 module number = 8

================================================================================================
========================
 Module number = 8
================================================================================================
========================
 [LC]   [    INST:SLI:MAC:GLSRC]   [IETH]    [ST]   <========>  [FM]   [    INST:SLI:MAC:GLSRC]
[IETH]   [ST]
```

```
================================================================================================================
 ...
[ 8]   [  1 :   1 :   9 :  0x0]   [iEth10]   [UP]  <========>  [22]   [  1 :   0 :   4 :
0x20]   [iEth35]   [UP]
```

我们看到插在插槽8中的线卡上的MAC标识符9映射到插在机箱插槽22中的交换矩阵模块。我们预计在内部链路iEth10上会看到CRC错误。我们可以使用**show system internal fabric connectivity stats module 8命令进行验**证。这里显示了一个示例。

```
Nexus9500# show system internal fabric connectivity stats module 8

Internal Link-info Stats Linecard slot:8
--------------------------------------------------------------------------------
 LC-Slot  LC-Unit   LC-iEthLink   MUX   FM-Slot   FM-Unit   FM-iEthLink    CRC
--------------------------------------------------------------------------------
   8        0     iEth01         -         22        0      iEth18        0
   8        0     iEth02         -         22        1      iEth50        0
   8        0     iEth03         -         23        0      iEth18        0
   8        0     iEth04         -         23        1      iEth50        0
   8        0     iEth05         -         24        0      iEth18        0
   8        0     iEth06         -         24        1      iEth50        0
   8        0     iEth07         -         26        0      iEth18        0
   8        0     iEth08         -         26        1      iEth50        0
   8        1     iEth09         -         22        0      iEth03        0
   8        1     iEth10         -         22        1      iEth35        1784603561
```

步骤2.将交换矩阵模块上的iEth链路映射到连接的线路卡并检查存储的CRC

接下来，我们按照与场景3相同的流程，检查接收CRC的iEth内部链路、这些CRC是否根据交换矩阵模块的ASIC进行堆叠，以及哪个线卡连接到交换矩阵模块的iEth内部链路。此处使用**show system internal fabric connectivity stats module {x}**命令、**show hardware internal errors module {x}**命令和**show system internal fabric link-state module {x}**命令显示了此示例。

```
Nexus9500# show system internal fabric connectivity stats module 22
Internal Link-info Stats Fabriccard slot:22
--------------------------------------------------------------------------------
 FM-Slot  FM-Unit   FM-iEthLink   LC-Slot   LC-Unit   LC-EthLink    MUX    CRC
   22         1      iEth56          7         3      iEth26         -     1171851894
Nexus9500# show hardware internal errors module 22 | i CRC|Stomp|Inst
Instance:1
196635 Interface Inbound Errors (CRC,len,Algn Err)   0000054593935847  27:0
1048603 Interface Inbound CRC Error Stomped           0000054593935846  27:0
Nexus9500# show system internal fabric link-state module 22 | i MAC|iEth56

 [FM]   [    INST:SLI:MAC:GLSRC]   [IETH]    [ST]   <========>  [LC]   [    INST:SLI:MAC:GLSRC]
[IETH]   [ST]
 [22]   [  1 :   4 :  27 : 0x18]   [iEth56]   [UP]  <========>  [ 7]   [  3 :   1 :   9 :
0x0]   [iEth26]   [UP]
```

步骤3.跟踪入口模块上带有无效CRC的帧的源

确定入口线卡（在此场景中，插入插槽7中的线卡通过iEth26连接到插入插槽22中的交换矩阵模块的iEth56）后，我们确定损坏的帧进入交换机的哪个入口端口。这通过**show interface counters errors module {x} non-zero**命令完成。show hardware internal errors module {x}命令和show interface hardware-mappings命令的输出可以验证收到的帧是无效还是存储的CRC。这里显示了一个示例，损坏的帧通过前面板接口Ethernet7/32进入交换机。

```
Nexus9500# show interface counters errors module 7 non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth7/32               0          0          0 4128770335          0           0

--------------------------------------------------------------------------------
Port         Stomped-CRC
--------------------------------------------------------------------------------
Eth7/32      4129998971
Nexus9500# show hardware internal errors module 7 | i i CRC|Stomp|Inst
<snip>
Instance:3
196619 Interface Inbound Errors (CRC,len,Algn Err)   0000054901402307  11:0
1048587 Interface Inbound CRC Error Stomped           0000054901402308  11:0
Nexus9500# show interface hardware-mappings | i Name|Eth7
<snip>
Name       Ifindex  Smod Unit HPort FPort NPort VPort Slice SPort SrcId MacId MacSP VIF  Block
BlkSrcID
...
Eth7/32    1a303e00 28   3    48    255   124   -1    1     8     16    11    0     125  0
16
```

场景4总结

我们已经确认，Nexus 9500在物理接口Ethernet7/32上接收带有堆积CRC的帧。这意味着Ethernet7/32链路远程端的设备正在堆积CRC这些帧；格式错误的帧的根源不是直接连接到Ethernet7/32的链路，而是更下游的链路。应对下游网络设备执行更多故障排除，以确定这些错误帧的来源。

相关信息

- 适用于低延迟环境的直通和存储转发以太网交换

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）。