

了解Nexus交换机上的循环冗余校验错误

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[适用硬件](#)

[CRC定义](#)

[CRC错误定义](#)

[CRC错误的常见症状](#)

[在Windows主机上收到错误](#)

[Linux主机上的RX错误](#)

[网络设备上的CRC错误](#)

[存储转发网络设备上的输入错误](#)

[直通网络设备的输入和输出错误](#)

[跟踪和隔离CRC错误](#)

[CRC错误的根本原因](#)

[解决CRC错误](#)

[相关信息](#)

简介

本文档介绍有关在接口计数器和Cisco Nexus交换机统计信息上观察到的循环冗余校验(CRC)错误的详细信息。

先决条件

要求

思科建议您了解以太网交换和Cisco NX-OS命令行界面(CLI)的基础知识。有关详细信息，请参阅以下适用文档之一：

- [Cisco Nexus 9000 NX-OS基础配置指南，版本10.2\(x\)](#)
- [Cisco Nexus 9000系列NX-OS基础配置指南，版本9.3\(x\)](#)
- [Cisco Nexus 9000系列NX-OS基础配置指南，版本9.2\(x\)](#)
- [Cisco Nexus 9000系列NX-OS基础配置指南，版本7.x](#)
- [以太网故障排除](#)

使用的组件

本文档中的信息基于以下软件和硬件版本：

- 从NX-OS软件版本9.3(8)开始的Nexus 9000系列交换机

- 从NX-OS软件版本9.3(8)开始的Nexus 3000系列交换机

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。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

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

本文档介绍有关在Cisco Nexus系列交换机的接口计数器上观察到的循环冗余校验(CRC)错误的详细信息。本文档介绍CRC是什么、在以太网帧的帧校验序列(FCS)字段中如何使用CRC、Nexus交换机上如何显示CRC错误、CRC错误在存储转发交换和直通交换场景中如何交互、CRC错误最可能的根本原因，以及如何排除和解决CRC错误。

适用硬件

本文档中的信息适用于所有Cisco Nexus系列交换机。本文档中的某些信息也适用于其他Cisco路由和交换平台，如Cisco Catalyst路由器和交换机。

CRC定义

CRC是计算机和存储网络中常用的一种错误检测机制，用于识别在传输过程中更改或损坏的数据。当连接到网络的设备需要传输数据时，该设备根据导致固定长度数的数据运行基于循环码的计算算法。此固定长度的数字称为CRC值，但通常称为CRC。此CRC值附加到数据，并通过网络传输到另一台设备。该远程设备对数据运行相同的循环编码算法，并将结果值与附加到数据的CRC进行比较。如果两个值都匹配，则远程设备会假设数据在网络中传输而未损坏。如果值不匹配，则远程设备会假设数据在通过网络传输时已损坏。此损坏的数据不能受信任，将被丢弃。

CRC用于跨多种计算机网络技术(如以太网（有线和无线变体）、令牌环、异步传输模式(ATM)和帧中继)进行错误检测。以太网帧在帧的末尾（紧接帧的负载后）有一个32位帧校验序列(FCS)字段，其中32位CRC值为的下界。

例如，假设两台名为Host-A和Host-B的主机通过其网络接口卡(NIC)直接连接。Host-A需要通过网络将句子“这是一个示例”发送到Host-B。主机A使用负载“这是一个示例”绘制发往主机B的以太网帧，并计算该帧的CRC值是十六进制值0xABCD。主机A将CRC值0xABCD插入以太网帧的FCS字段，然后将以太网帧从主机A的网卡传输到主机B。

主机B收到此帧时，将使用与主机A完全相同的算法计算该帧的CRC值。主机B计算出帧的CRC值是十六进制值0xABCD，这向主机B表示在将帧传输到主机B时以太网帧未损坏。

CRC错误定义

当设备（网络设备或连接到网络的主机）在帧的FCS字段中收到一个以太网帧，该帧的CRC值与设备为该帧计算的CRC值不匹配时，会发生CRC错误。

最好通过一个例子来说明这一概念。假设两台名为Host-A和Host-B的主机通过其网络接口卡(NIC)直接相连。主机A需要通过网络将句子“这是一个示例”发送到主机B。主机A使用负载“这是一个示例

”绘制发往主机B的以太网帧，并计算该帧的CRC值是十六进制值0xABCD。主机A将CRC值0xABCD插入以太网帧的FCS字段，然后将以太网帧从主机A的网卡传输到主机B。

但是，将主机A连接到主机B的物理介质损坏会损坏帧的内容，使得帧中的句子更改为“这是一个示例”，而不是期望的负载“这是一个示例”。

当主机B收到此帧时，它将计算该帧的CRC值，包括损坏的负载。主机B计算出帧的CRC值是十六进制值0xDEAD，与以太网帧的FCS字段中的0xABCD CRC值不同。CRC值的这种差异告诉主机B，在将该帧传输到主机B时，以太网帧已损坏。因此，主机B无法信任该帧此以太网帧的内容，因此它会丢弃它。主机B通常会在其网络接口卡(NIC)上增加某种错误计数器，例如“输入错误”、“CRC错误”或“RX错误”计数器。

CRC错误的常见症状

CRC错误通常通过以下两种方式之一出现：

1. 网络连接设备接口上的递增或非零错误计数器。
2. 因网络连接设备丢弃损坏的帧而流经网络的流量的数据包/帧丢失。

这些错误的表现方式稍有不同，具体取决于您使用的设备。这些子部分将详细介绍每种类型的设备。

在Windows主机上收到错误

Windows主机上的CRC错误通常显示为 `netstat -e` 命令的输出中显示的非零 **Received Errors** 计数器。Windows主机的命令提示符下的非零 **Received Errors** 计数器示例如下：

```
>netstat -e
Interface Statistics


```

	Received	Sent
Bytes	1116139893	3374201234
Unicast packets	101276400	49751195
Non-unicast packets	0	0
Discards	0	0
Errors	47294	0
Unknown protocols	0	

NIC及其相应的驱动程序必须支持对NIC接收的CRC错误进行记帐，以使 `netstat -e` 命令报告的“已接收错误”数准确。大多数现代网卡及其各自的驱动程序都支持准确统计网卡收到的CRC错误。

Linux主机上的RX错误

Linux主机上的CRC错误通常以非零“RX错误”计数器的形式显示在 `ifconfig` 命令的输出中。来自Linux主机的非零RX错误计数器的示例如下：

```
$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.0.2.10 netmask 255.255.255.128 broadcast 192.0.2.255
    inet6 fe80::10 prefixlen 64 scopeid 0x20<link>
    ether 08:62:66:be:48:9b txqueuelen 1000 (Ethernet)
    RX packets 591511682 bytes 214790684016 (200.0 GiB)
    RX errors 478920 dropped 0 overruns 0 frame 0
```

```
TX packets 85495109 bytes 288004112030 (268.2 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Linux主机上的CRC错误也可以显示为ip -s link show命令输出中显示的非零“RX错误”计数器。以下是来自Linux主机的非零RX错误计数器的示例：

```
$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
    link/ether 08:62:66:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    32246366102 444908978 478920      647      0      419445867
    TX: bytes  packets  errors  dropped carrier collsns
    3352693923 30185715 0        0        0        0
    altname enp11s0
```

为使ifconfig或ip -s link show命令报告的RX错误数准确，网卡及其相应驱动程序必须支持对网卡收到的CRC错误进行记帐。大多数现代网卡及其各自的驱动程序都支持准确统计网卡收到的CRC错误。

网络设备上的CRC错误

网络设备在两种转发模式之一下运行 — 存储转发模式和直通转发模式。网络设备处理收到的CRC错误的方式因转发模式而异。此处的子部分将描述每种转发模式的具体行为。

存储转发网络设备上的输入错误

当以存储转发模式运行的网络设备收到帧时，网络设备将在验证帧的CRC值之前缓冲整个帧（“存储”），对该帧做出转发决定，并将帧从接口（“转发”）传出。因此，当以存储转发模式运行的网络设备在特定接口上收到CRC值不正确的损坏帧时，它会丢弃该帧并增加接口上的“输入错误”计数器。

换句话说，损坏的以太网帧不会由以存储转发模式运行的网络设备转发；它们会被丢弃在入口。

Cisco Nexus 7000和7700系列交换机在存储转发模式下运行。Nexus 7000或7700系列交换机的非零输入错误计数器和非零CRC/FCS计数器的示例如下：

```
switch# show interface
<snip>
Ethernet1/1 is up
RX
 241052345 unicast packets  5236252 multicast packets  5 broadcast packets
 245794858 input packets  17901276787 bytes
 0 jumbo packets  0 storm suppression packets
 0 runts  0 giants  579204 CRC/FCS  0 no buffer
 579204 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
```

CRC错误也可以在show interface counters错误输出中显示为非零“FCS-Err”计数器。此命令输出中的“Rcv-Err”计数器也将具有非零值，该值是接口接收的所有输入错误（CRC或其他）的总和。相应示例如下：

```
switch# show interface counters errors
<snip>
```

Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize	OutDiscards
Eth1/1	0	579204	0	579204	0	0

直通网络设备的输入和输出错误

当以直通转发模式运行的网络设备开始接收帧时，网络设备将对帧的报头做出转发决定，并在收到足够的帧后立即开始从接口发送帧以作出有效的转发决定。由于帧和数据包报头位于帧的开头，因此通常在收到帧的负载之前做出此转发决定。

以太网帧的FCS字段位于帧的末尾，紧挨着帧的负载。因此，在直通转发模式下运行的网络设备在计算帧的CRC时已经开始从另一个接口发送帧。如果网络设备计算的帧的CRC与FCS字段中的CRC值不匹配，这意味着网络设备将损坏的帧转发到网络。发生这种情况时，网络设备将增加两个计数器：

1. 最初接收损坏帧的接口上的“输入错误”计数器。
2. 传输损坏帧的所有接口上的“输出错误”计数器。对于单播流量，这通常是单个接口 — 但是，对于广播、组播或未知单播流量，这可能是一个或多个接口。

此处显示了一个示例，其中**show interface**命令的输出表示在网络设备的Ethernet1/1上收到了多个损坏的帧，并且由于网络设备的直通转发模式从Ethernet1/2传输出去：

```
switch# show interface
<snip>
Ethernet1/1 is up
RX
 46739903 unicast packets 29596632 multicast packets 0 broadcast packets
 76336535 input packets 6743810714 bytes
 15 jumbo packets 0 storm suppression bytes
 0 runts 0 giants 47294 CRC 0 no buffer
 47294 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

Ethernet1/2 is up
TX
 46091721 unicast packets 2852390 multicast packets 102619 broadcast packets
 49046730 output packets 3859955290 bytes
 50230 jumbo packets
 47294 output error 0 collision 0 deferred 0 late collision
 0 lost carrier 0 no carrier 0 babble 0 output discard
 0 Tx pause
```

CRC错误也可以在入口接口上显示为非零“FCS-Err”计数器，在**show interface counters**错误输出中，在出口接口上显示为非零“Xmit-Err”计数器。此命令输出中入口接口上的“Rcv-Err”计数器也将具有非零值，该值是接口接收的所有输入错误（CRC或其他）的总和。相应示例如下：

```
switch# show interface counters errors
<snip>
-----
Port          Align-Err  FCS-Err  Xmit-Err  Rcv-Err  UnderSize  OutDiscards
-----
Eth1/1        0          47294    0          47294    0           0
Eth1/2        0           0        47294    0           0           0
```

网络设备还将以特定方式修改帧的FCS字段中的CRC值，以向上游网络设备表示此帧已损坏。此行为称为“自定义”CRC。修改CRC的精确方式因平台而异，但通常涉及反转帧的FCS字段中当前的CRC值。以下是一个示例：

Original CRC: 0xABCD (1010101111001101)

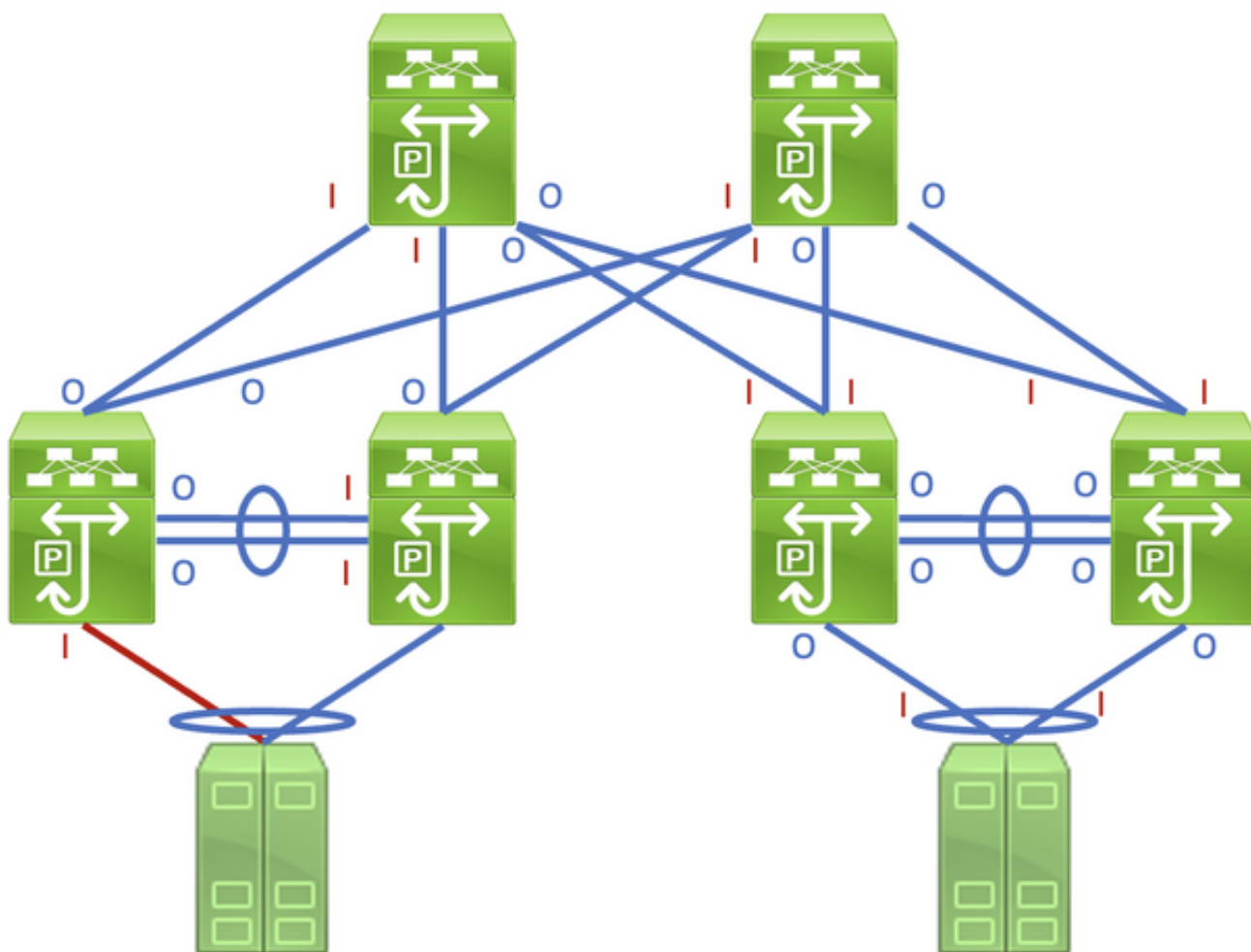
Stomped CRC: 0x5432 (0101010000110010)

由于此行为，以直通转发模式运行的网络设备可以在整个网络中传播损坏的帧。如果网络由多个以直通转发模式运行的网络设备组成，则单个损坏的帧可能导致网络内多个网络设备上的输入错误和输出错误计数器增加。

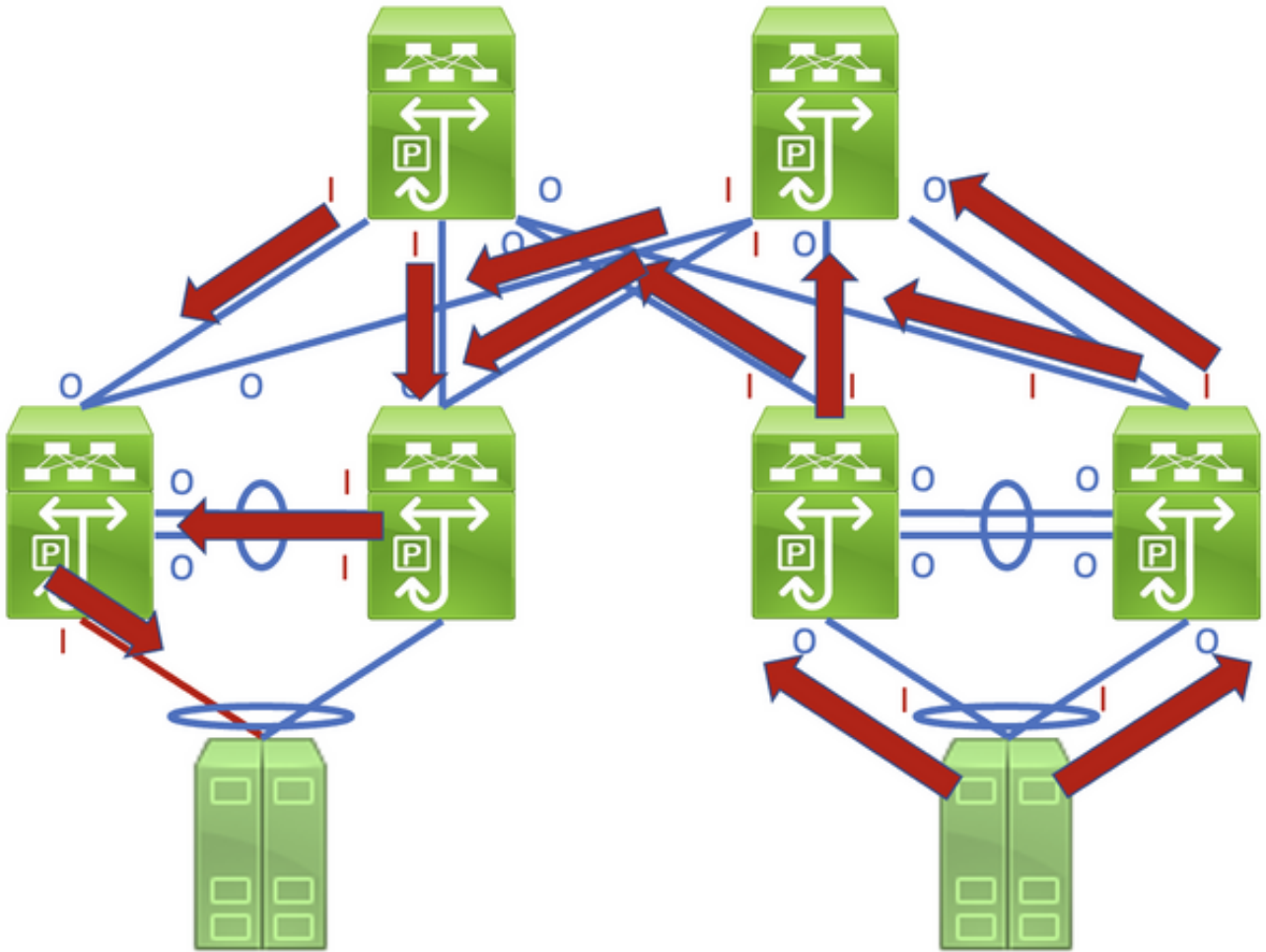
跟踪和隔离CRC错误

要识别并解决CRC错误的根本原因，第一步是将CRC错误的来源隔离到网络中两台设备之间的特定链路。连接到此链路的一台设备的接口输出错误计数器的值为零或未递增，而连接到此链路的另一台设备的接口输入错误计数器将非零或递增。这表明，传输到远程设备时，一个设备接口的流量将完好无损地传出，并被链路上另一设备的入口接口视为输入错误。

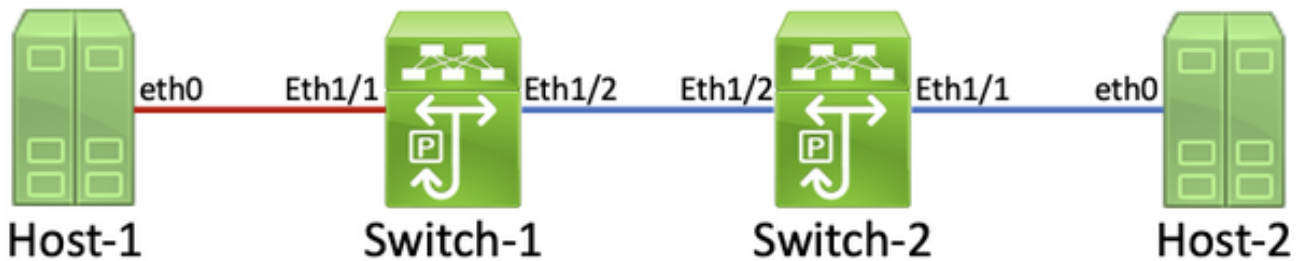
在由以存储转发模式运行的网络设备组成的网络中识别此链路是一项简单的任务。但是，在由以直通转发模式运行的网络设备组成的网络中识别此链路比较困难，因为许多网络设备将具有非零输入和输出错误计数器。此处的拓扑中可以看到这种现象的一个示例，其中以红色突出显示的链路被损坏，以致通过链路的流量被损坏。标有红色“1”的接口表示可能存在非零输入错误的接口，而标有蓝色“0”的接口表示可能存在非零输出错误的接口。



识别故障链路需要您通过非零输入和输出错误计数器递归地跟踪网络中跟随的“路径”损坏帧，非零输入错误指向网络中损坏链路的上游。这在图中演示。



最好通过一个示例演示跟踪和识别损坏链路的详细过程。请考虑以下拓扑：



在此拓扑中，名为Switch-1的Nexus交换机的接口Ethernet1/1通过Host-1的网络接口卡(NIC)eth0连接到名为Host-1的主机。Switch-1的接口Ethernet1/2通过Switch-2的接口Ethernet1/2连接到另一台名为Switch-2的Nexus交换机。接口Ethernet1/1交换机2的NIC eth0连接到名为Host-2的主机。

通过Switch-1的Ethernet1/1接口的Host-1和Switch-1之间的链路损坏，导致通过链路的流量间歇性损坏。但是，我们尚不知道此链路是否损坏。我们必须通过非零或递增输入和输出错误计数器跟踪损坏帧在网络中离开的路径，以查找此网络中损坏的链路。

在本例中，Host-2的NIC报告它正在接收CRC错误。

```
Host-2$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
    link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
```

```

32246366102 444908978 478920          647      0          419445867
TX: bytes  packets  errors  dropped carrier collsns
3352693923 30185715 0        0         0         0
altname enp11s0

```

您知道，Host-2的NIC通过接口Ethernet1/1连接到Switch-2。您可以使用**show interface**命令确认接口Ethernet1/1的输出错误计数器为非零。

```

Switch-2# show interface
<snip>
Ethernet1/1 is up
admin state is up, Dedicated Interface
  RX
    30184570 unicast packets  872 multicast packets  273 broadcast packets
    30185715 input packets  3352693923 bytes
    0 jumbo packets  0 storm suppression bytes
    0 runs  0 giants  0 CRC  0 no buffer
    0 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
  TX
    444907944 unicast packets  932 multicast packets  102 broadcast packets
    444908978 output packets  32246366102 bytes
    0 jumbo packets
    478920 output error  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble  0 output discard
    0 Tx pause

```

由于接口Ethernet1/1的输出错误计数器为非零值，因此Switch-2的另一个接口很可能具有非零输入错误计数器。您可以使用**show interface counters errors non-zero**命令来确定Switch-2的任何接口是否具有非零输入错误计数器。

```

Switch-2# show interface counters errors non-zero
<snip>
-----
Port                Align-Err    FCS-Err    Xmit-Err    Rcv-Err    UnderSize  OutDiscards
-----
Eth1/1                0             0          478920         0             0             0
Eth1/2                0           478920         0          478920         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
-----

```

您可以看到Switch-2的Ethernet1/2有一个非零输入错误计数器。这表明交换机2在此接口上收到损坏的流量。您可以通过思科发现协议(CDP)或链路本地发现协议(LLDP)功能确认哪台设备连接到交换机2的以太网1/2。此处使用**show cdp neighbors**命令显示了此示例。

```

Switch-2# show cdp neighbors

```



```
<snip>
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
S - Switch, H - Host, I - IGMP, r - Repeater,
V - VoIP-Phone, D - Remotely-Managed-Device,
s - Supports-STP-Dispute
```

```
Device-ID          Local Intrfce  Hldtme  Capability  Platform          Port ID
Switch-1(FD012345678)
                   Eth1/2        125     R S I s     N9K-C93180YC-    Eth1/2
```

您现在知道，Switch-2的Ethernet1/2接口上收到来自Switch-1的Ethernet1/2接口的损坏流量，但您不知道Switch-1的Ethernet1/2和Switch-2的Ethernet1/2之间的链路是否损坏并导致损坏，或者Switch-1是直通交换机转发其收到的损坏流量。您必须登录Switch-1以验证这一点。

您可以使用show interfaces命令确认Switch-1的Ethernet1/2接口具有非零输出错误计数器。

```
Switch-1# show interface
<snip>
Ethernet1/2 is up
admin state is up, Dedicated Interface
RX
 30581666 unicast packets  178 multicast packets  931 broadcast packets
30582775 input packets  3352693923 bytes
0 jumbo packets  0 storm suppression bytes
0 runs  0 giants  0 CRC  0 no buffer
0 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
TX
454301132 unicast packets  734 multicast packets  72 broadcast packets
454301938 output packets  32246366102 bytes
0 jumbo packets
478920 output error  0 collision  0 deferred  0 late collision
0 lost carrier  0 no carrier  0 babble  0 output discard
0 Tx pause
```

您可以看到Switch-1的Ethernet1/2有一个非零输出错误计数器。这表明Switch-1的Ethernet1/2和Switch-2的Ethernet1/2之间的链路没有损坏 — 相反，Switch-1是转发其在其他接口上收到的损坏流量的直通交换机。如前在Switch-2中演示的，您可以使用show interface counters errors non-zero命令来确定Switch-1的任何接口是否具有非零输入错误计数器。

```
Switch-1# show interface counters errors non-zero
<snip>
-----
Port          Align-Err    FCS-Err    Xmit-Err    Rcv-Err    UnderSize  OutDiscards
-----
Eth1/1        0           478920     0           478920     0           0
Eth1/2        0           0          478920     0           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

您可以看到Switch-1的Ethernet1/1有一个非零输入错误计数器。这表明交换机1正在此接口上接收损坏的流量。我们知道此接口连接到Host-1的eth0 NIC。我们可以查看Host-1的eth0 NIC接口统计信息，以确认Host-1是否从此接口发送损坏的帧。

```
Host-1$ ip -s link show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
    link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped  overrun  mcast
    73146816142 423112898 0        0        0        437368817
    TX: bytes  packets  errors  dropped  carrier  collsns
    3312398924 37942624 0        0        0        0
    altname enp11s0
```

Host-1的eth0 NIC统计信息表明主机未传输损坏的流量。这表明主机1的eth0和交换机1的以太网接口1/1之间的链路已损坏，是此流量损坏的源。需要对此链路执行进一步的故障排除，以确定导致此损坏的故障组件并更换它。

CRC错误的根本原因

CRC错误最常见的根本原因是两台设备之间物理链路的损坏或故障组件。示例包括：

- 物理介质（铜缆或光纤）或直连电缆(DAC)出现故障或损坏。
- 收发器/光纤出现故障或损坏。
- 配线面板端口故障或损坏。
- 网络设备硬件故障（包括特定端口、线卡专用集成电路[ASIC]、介质访问控制[MAC]、交换矩阵模块等），
- 主机中插入的网络接口卡出现故障。

一个或多个配置错误的设备也可能在网络中无意中导致CRC错误。其中一个示例是网络内两个或多个设备之间的最大传输单元(MTU)配置不匹配，导致大数据包被错误截断。识别并解决此配置问题也可以纠正网络中的CRC错误。

解决CRC错误

您可以通过消除流程识别特定故障组件：

1. 使用相同类型的已知良好的物理介质更换物理介质（铜或光纤）或DAC。
2. 将插入设备接口的收发器更换为同一型号的已知良好的收发器。如果这无法解决CRC错误，请用同一型号的已知良好的收发器替换插入其他设备接口的收发器。
3. 如果任何配线面板用作损坏链路的一部分，请将链路移到配线面板上已知良好的端口。或者，如果可能，无需使用配线面板即可连接链路，从而消除配线面板作为潜在根本原因的可能。
4. 将损坏的链路移到每台设备上另一个已知良好的端口。您需要测试多个不同的端口以隔离MAC、ASIC或线卡故障。
5. 如果损坏的链路涉及主机，请将链路移到主机上的其他网卡。或者，将损坏的链路连接到已知良好的主机，以隔离主机网卡的故障。

如果故障组件是受有效支持合同覆盖的思科产品 ([如思科网络设备或收发器](#))，您可以向[Cisco TAC](#)提交支持案例，详细说明通过退货授权(RMA)更换故障组件的故障排除。

相关信息

- [Nexus 9000云扩展ASIC CRC识别和跟踪程序](#)
- [技术支持和文档 - Cisco Systems](#)