

使用Nexus平台上的接口计数器排除丢包故障

目录

- [简介](#)
- [先决条件](#)
- [要求](#)
- [使用的组件](#)
- [拓扑](#)
- [背景](#)
- [标识接口](#)
 - [N9K1中的路由](#)
 - [N9K2中的路由](#)
 - [N9K3中的路由](#)
- [确定数据包大小](#)
 - [执行测试](#)
 - [检验ICMP请求](#)
 - [检验ICMP应答](#)

简介

本文档介绍如何使用Nexus接口计数器排除数据包丢失故障。

先决条件

要求

Cisco 建议您了解以下主题：

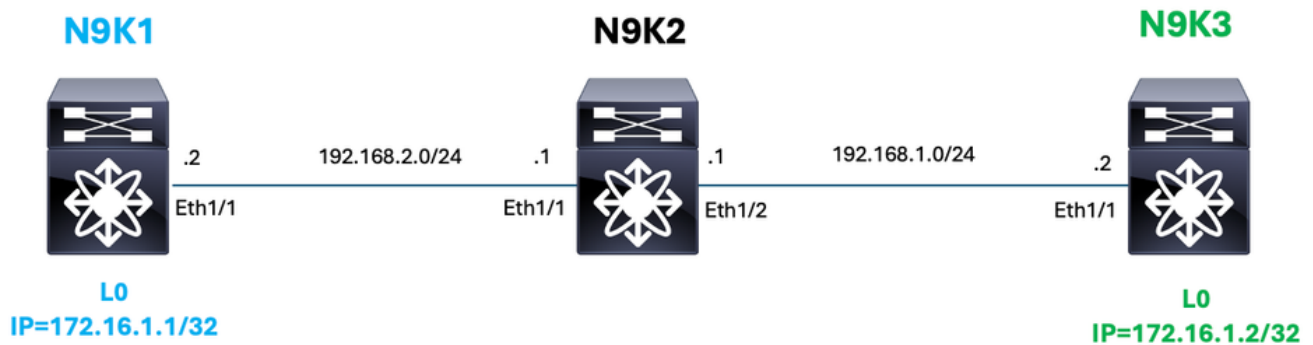
- NXOS平台

使用的组件

名称	平台	version
N9K1	N9K-C93108TC-EX	9.3(10)
N9K2	N9K-C93108TC-EX	9.3(10)
N9K3	N9K-C93108TC-EX	9.3(10)

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。

拓扑



背景

在某些环境中，传统的数据包捕获方法（如ELAM或SPAN）不能作为诊断网络问题的可行选项。但是，Nexus接口数据包计数器为排除数据包丢弃故障提供了有价值的替代方法。需要注意的是，特定计数器的可用性可能会因网络配置而异，因此这种故障排除方法不能普遍适用。

在本示例中，演示了如何使用Nexus接口计数器来排除N9K1(172.16.1.1)和N9K3(172.16.1.2)的环回接口之间的连接问题。

标识接口

对于每个设备，必须标识入口和出口接口，以便通过此示例命令标识此接口：使用show ip route。

N9K1中的路由

```
<#root>
```

N9K1

```
# sh ip route 172.16.1.2
<Snipped>
172.16.1.2/32, ubest/mbest: 1/0
    *via 192.168.2.1,
```

Eth1/1

```
, [1/0], static
```

对于nexus N9K1，使用接口Eth1/1。

N9K2中的路由

```
<#root>
```

N9K2

```
# sh ip route 172.16.1.1
<Snipped>
172.16.1.1/32, ubest/mbest: 1/0 time
    *via 192.168.2.2,
```

Eth1/1

, [1/0], static

N9K2

```
# sh ip route 172.16.1.2
<Snipped>
172.16.1.2/32, ubest/mbest: 1/0 time
    *via 192.168.1.2,
```

Eth1/2

, [1/0], static

对于nexus N9K1，使用接口Eth1/1和Eth1/2。

N9K3中的路由

<#root>

N9K3

```
# sh ip route 172.16.1.1
<Snipped>
172.16.1.1/32, ubest/mbest: 1/0 time
    *via 192.168.1.1,
```

Eth1/1

, [1/0], static

对于nexus N9K1，使用接口Eth1/1。

确定数据包大小

为了使用接口计数器排除数据包丢弃故障，需要确定一个未增加的计数器。

在下一个示例中，运行sh interface e1/1 counter detailed命令两次，如果可以观察到从512到1023字节的计数器Packets对RX和TX没有增加。

此过程需要在源和目标之间涉及的所有设备上完成。

<#root>

N9K1# sh interface e1/1 counters detailed

Ethernet1/1

Rx Packets: 31774
Rx Unicast Packets: 8419
Rx Multicast Packets: 23784
Rx Broadcast Packets: 3
Rx Bytes: 8115383
Rx Packets from 0 to 64 bytes: 322
Rx Packets from 65 to 127 bytes: 22822
Rx Packets from 128 to 255 bytes: 3393
Rx Packets from 256 to 511 bytes: 1652

Rx Packets from 512 to 1023 bytes: 63

Rx Packets from 1024 to 1518 bytes: 3522
Tx Packets: 26430
Tx Unicast Packets: 7351
Tx Multicast Packets: 19509
Tx Broadcast Packets: 2
Tx Bytes: 5114894
Tx Packets from 0 to 64 bytes: 90
Tx Packets from 65 to 127 bytes: 20724
Tx Packets from 128 to 255 bytes: 2243
Tx Packets from 256 to 511 bytes: 1642

Tx Packets from 512 to 1023 bytes: 10

Tx Packets from 1024 to 1518 bytes: 1766

N9K1# sh interface e1/1 counters detailed

Ethernet1/1

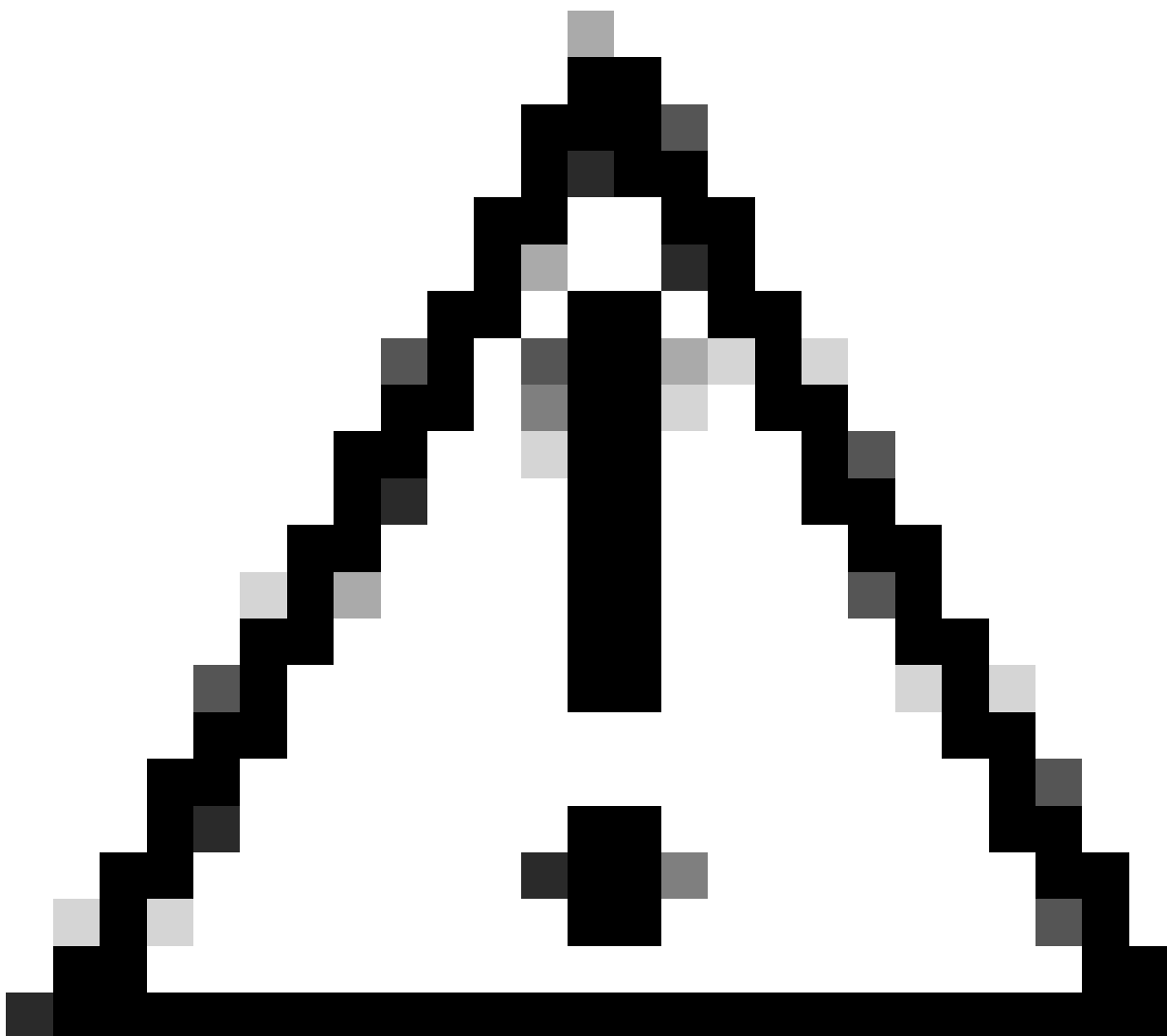
Rx Packets: 31821
Rx Unicast Packets: 8437
Rx Multicast Packets: 23817
Rx Broadcast Packets: 3
Rx Bytes: 8125733
Rx Packets from 0 to 64 bytes: 329
Rx Packets from 65 to 127 bytes: 22878
Rx Packets from 128 to 255 bytes: 3468
Rx Packets from 256 to 511 bytes: 1670

Rx Packets from 512 to 1023 bytes: 63

Rx Packets from 1024 to 1518 bytes: 3544
Tx Packets: 26467
Tx Unicast Packets: 7367
Tx Multicast Packets: 19534
Tx Broadcast Packets: 2
Tx Bytes: 5121572
Tx Packets from 0 to 64 bytes: 95
Tx Packets from 65 to 127 bytes: 20768
Tx Packets from 128 to 255 bytes: 2290
Tx Packets from 256 to 511 bytes: 1657

Tx Packets from 512 to 1023 bytes: 10

Tx Packets from 1024 to 1518 bytes: 1798



警告：在生产环境中，可以清除接口计数器，以便确定哪个计数器没有增加。对于MTU设置为最大值的接口，可以找到大于1518的计数器。如果具有特定大小的数据包未通过nexus，则不会显示计数器。

执行测试

对于此测试，由于使用受控环境，因此所有设备都使用计数器Packets from 1024 to 1518。所有接口的计数器在测试之前被清除：

```
<#root>
```

```
N9K1
```

```
# clear counters interface
```

```
e1/1
```

```
N9K2
```

```
# clear counters interface
```

```
e1/1-2
```

```
N9K3
```

```
# clear counters interface
```

```
e1/1
```

在所有nexus中，可以运行下一命令来验证没有具有所需数据包大小的流量通过nexus;人们希望看到什么；

```
<#root>
```

```
N9K1
```

```
# sh int
```

```
e1/1
```

```
cou detailed | i i " 1024 to 1518"
```

```
N9K2
```

```
# sh int
```

```
e1/1-2
```

```
cou detailed | i i " 1024 to 1518"
```

```
N9K3
```

```
# sh int
```

```
e1/1
```

```
cou detailed | i i " 1024 to 1518"
```

现在所有计数器都已清除，可以生成ping，指定数据包大小介于1024-1518之间，并设置DF-BIT。

```
<#root>
```

```
N9K1
```

```
# ping
```

```
172.16.1.2
```

```
source 172.16.1.1 packet-size 1050 df-bit
PING 172.16.1.2(172.16.1.2) from 172.16.1.1: 1050 data bytes
1058 bytes from 172.16.1.2: icmp_seq=0 ttl=254 time=1.102 ms
1058 bytes from 172.16.1.2: icmp_seq=1 ttl=254 time=0.668 ms
1058 bytes from 172.16.1.2: icmp_seq=2 ttl=254 time=0.644 ms
1058 bytes from 172.16.1.2: icmp_seq=3 ttl=254 time=0.626 ms
1058 bytes from 172.16.1.2: icmp_seq=4 ttl=254 time=0.631 ms
```

```
--- 172.16.1.2 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0.00% packet loss
```

round-trip min/avg/max = 0.626/0.734/1.102 ms

检验ICMP请求

在下一个示例中，您可以观察从N9K1到N9K3的ICMP请求的相关设备上的计数器如何在TX/RX方向上增加。

N9K1	N9K2
<pre><#root> N9K1# sh int e1/1 cou detailed i i " 1024 to 1518" 5 Rx Packets from 1024 to 1518 bytes: 0 Tx Packets from 1024 to 1518 bytes: 5</pre>	<pre><#root> N9K2# sh int e1/1 cou detailed i i " 1024 to 1518" 5 Rx Packets from 1024 to 1518 bytes: 5 Tx Packets from 1024 to 1518 bytes: 0 N9K2# sh int e1/2 cou detailed i i " 1024 to 1518" 5 Rx Packets from 1024 to 1518 bytes: 0 Tx Packets from 1024 to 1518 bytes: 5</pre>
可以看到N9K1在接口e1/1上发送了5个数据包	可以看到N9K2在接口e1/1上接收了5个数据包，并在接口e1/2上发送了5个数据包

检验ICMP应答

验证ICMP请求补丁后，您可以继续查看ICMP应答。

在下一个示例中，您可以观察从N9K3到N9K1的ICMP应答的相关设备上的计数器如何在TX/RX方向上增加

N9K1	N9K2
<pre><#root> N9K1# sh int e1/1 cou detailed i i " 1024 to 1518" Rx Packets from 1024 to 1518 bytes: 5 Tx Packets from 1024 to 1518 bytes: 5</pre>	<pre><#root> N9K2# sh int e1/1 cou detailed i i " 1024 to 1518" Rx Packets from 1024 to 1518 bytes: 5 Tx Packets from 1024 to 1518 bytes: 5 N9K2# sh int e1/2 cou detailed i i " 1024 to 1518" Rx Packets from 1024 to 1518 bytes: 5 Tx Packets from 1024 to 1518 bytes: 5</pre>
可以看到N9K1在接口e1/1上收到5个数据包	可以看到N9K2在接口e1/1上发送了5个数据包，在接口e1/2上接收了5个数据包

通过此测试，可以确认流量在3台交换机之间正确流动。如果其中一个nexus的计数器存在差异，则表示可以丢弃流量的RX或TX。

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）。