

排除Cisco Nexus 9000上损坏的以太网数据包故障

目录

[简介](#)

[背景信息](#)

[交换机如何处理数据包](#)

[流量通过N9K时用标记VLAN修改填充](#)

[解决方案](#)

简介

本文档介绍当填充信息损坏或格式错误时如何对Cisco Nexus 9000上损坏的以太网数据包进行故障排除。

背景信息

以太网帧的最小大小为64字节，无论是否存在VLAN标记。

最小以太网负载大小为：

- 46字节（如果VLAN标记不存在）。
- 42字节（如果存在VLAN标记）。

您可以验证以下事实：

- 在维基百科上，节负载：https://en.wikipedia.org/wiki/Ethernet_frame
- 在IEEE 802.3标准

(http://people.ee.duke.edu/~mbrooke/EE164.02/Spring_2004/group_2/index_files/8023.pdf)中，MAC帧格式（无VLAN）在第39页第3.1.1节中定义，标记MAC帧的元素在第43页第3.5节中定义。

无论VLAN报头是否存在，以太网数据包的最小大小都为64字节。允许服务器发送包含VLAN的64字节长数据包，您应该接受该数据包并正确处理该数据包。

注意：此行为由Catalyst 4500x正确处理，而不是由Nexus 9k处理。

交换机如何处理数据包

步骤1.接收一个有效的64字节以太网帧。

步骤2.删除帧校验序列(FCS)，使数据包变长60字节。

步骤3.删除VLAN标记，使数据包长度变为56字节。

步骤4.添加填充以使数据包长度为60字节。

步骤5.添加FCS，使数据包长64字节。

当数据包通过直通交换机时，不应修改填充。

流量通过N9K时用标记VLAN修改填充

数据包不使用零填充，而是使用垃圾字符填充，在大多数情况下，由于校验和未修改，因此没有人使用这些数据，因此它没有影响。但是，如果客户有特殊用途，需要重新计算校验和，这些垃圾数据会导致最终校验和损坏（其他设备，如NAT/负载均衡器，也可能会发现问题）。

设备是N9K 93120TX（最初在9372TX上检测到），版本是最新的NXOS 7.0(3)I2(2a)。

在此处（1000base-T链路）使用硬件直接连接到N9K（无任何虚拟化）的Linux主机。

使用此配置：

```
interface Ethernet1/59
    switchport mode trunk
!
interface Ethernet1/60
    switchport mode trunk
```

linux configurations:

```
inet 10.2.1.1/24 brd 10.2.1.255 scope global eth1 <= native vlan
inet 10.1.1.1/24 brd 10.1.1.255 scope global eth1.100 <= taggued vlan 100
```

或

只需连接Windows主机并发送标记帧，它们就会触发问题。此外，确保网络接口卡(NIC)能够标记数据包。

交换机将非零填充添加到通过的帧。

例如:主机(Host)- [中继] N9K [中继] — 主机

您可以使用netcat发送和接收数据包。

如图所示，它在交换机上发送端（VLAN 100标记）端口e1/59。

```
6: eth1.100@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue state UP group default
link/ether 44:a8:42:2c:5f:c4 brd ff:ff:ff:ff:ff:ff
inet 10.1.1.1/24 brd 10.1.1.255 scope global eth1.100
valid_lft forever preferred_lft forever
inet6 fe80::46a8:42ff:fe2c:5fc4/64 scope link
valid_lft forever preferred_lft forever
```

```
root@s35-c2-0:~# nc 10.1.1.2 3002 -u
a
^C
root@s35-c2-0:~#
```

它在交换机上接收端 (VLAN 100标记)、端口e1/60 , 如图所示 :

```
7: eth1.100@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue state UP group default
link/ether 44:a8:42:2c:63:d1 brd ff:ff:ff:ff:ff:ff
inet 10.1.1.2/24 brd 10.1.1.255 scope global eth1.100
valid_lft forever preferred_lft forever
inet6 fe80::46a8:42ff:fe2c:63d1/64 scope link
valid_lft forever preferred_lft forever
```

```
root@s35-c2:~# nc -l -u -p 3002
a
^C
root@s35-c2:~#
```

如图所示 , 数据包被传输。

```
root@s35-c2-0:~# tcpdump -i eth1.100 -nvex
tcpdump: listening on eth1.100, link-type EN10MB (Ethernet), capture size 65535 bytes
10:42:20.953994 44:a8:42:2c:5f:c4 > 44:a8:42:2c:63:d1, ethertype IPv4 (0x0800), length 44: (tos 0x0, ttl 64, id 64283, offset 0, flags [DF], proto UDP (17), length 30)
  10.1.1.1.41675 > 10.1.1.2.3002: UDP, length 2
  0x0000: 4500 001e fb1b 4000 4011 29af 0a01 0101
  0x0010: 0a01 0102 a2cb 0bba 000a 1620 610a
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
root@s35-c2-0:~#
```

收到数据包 , 如图所示 :

```
10:43:12.665897 44:a8:42:2c:5f:c4 > 44:a8:42:2c:63:d1, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64, id 64283, offset 0, flags [DF], proto UDP (17), length 30)
  10.1.1.1.41675 > 10.1.1.2.3002: UDP, length 2
  0x0000: 4500 001e fb1b 4000 4011 29af 0a01 0101
  0x0010: 0a01 0102 a2cb 0bba 000a da45 610a 0000
  0x0020: 0000 0000 0000 0000 0000 7562 710e
^C
7 packets captured
7 packets received by filter
0 packets dropped by kernel
root@s35-c2:~#
```

如图所示 , 突出显示了错误的填充。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.1	10.1.1.2	UDP	60	Source port: 40849 Destination port: 3002

▶ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

▼ Ethernet II, Src: Dell_2c:5f:c4 (44:a8:42:2c:5f:c4), Dst: Dell_2c:63:d1 (44:a8:42:2c:63:d1)

- ▶ Destination: Dell_2c:63:d1 (44:a8:42:2c:63:d1)
- ▶ Source: Dell_2c:5f:c4 (44:a8:42:2c:5f:c4)
- Type: IP (0x0800)

Padding: 000000000000000000000000f1b7bc5c

▼ Internet Protocol Version 4, Src: 10.1.1.1 (10.1.1.1), Dst: 10.1.1.2 (10.1.1.2)

- 0100 = Version: 4
- 0101 = Header Length: 20 bytes
- ▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
- Total Length: 30
- Identification: 0xfb1d (64285)
- ▶ Flags: 0x02 (Don't Fragment)
- Fragment offset: 0
- Time to live: 64
- Protocol: UDP (17)
- ▶ Header checksum: 0x29ad [validation disabled]
- Source: 10.1.1.1 (10.1.1.1)
- Destination: 10.1.1.2 (10.1.1.2)
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]

▼ User Datagram Protocol, Src Port: 40849 (40849), Dst Port: 3002 (3002)

- Source Port: 40849 (40849)
- Destination Port: 3002 (3002)
- Length: 10
- ▼ Checksum: 0xdd7f [validation disabled]
- [Good Checksum: False]
- [Bad Checksum: False]
- [Stream index: 0]

▼ Data (2 bytes)

- Data: 610a
- [Length: 2]

0000	44 a8 42 2c 63 d1 44 a8 42 2c 5f c4 08 00 45 00	D.B,c.D. B,....E.
0010	00 1e fb 1d 40 00 40 11 29 ad 0a 01 01 01 0a 01	...@.@.).....
0020	01 02 9f 91 0b ba 00 0a dd 7f 61 0a 00 00 00 00a.....
0030	00 00 00 00 00 00 00 00 f1 b7 bc 5c\

这也与数据包分析器一起显示（在另一个数据包中，数据与之前的屏幕截图不同，但测试和Bug相同），

解决方案

解决方法是在连接了[此服务器](#)的接口上禁用缓冲增强。

```
C9396PX-1(config)# int et 1/7
C9396PX-1(config-if)# no buffer-boost
```

相关缺陷：

[CSCva46849](#) 60字节帧，带N9k上的dot1q报头L2交换