Catalyst 9000 系列交换机 MTU 故障排除

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

- MTU摘要表
- MTU问答
- 以太网帧

配置和验证MTU

- 配置MTU
- 验证MTU

排除MTU故障

- 拓扑
- 入口数据包丢弃（入口MTU更低）

配置和验证IP MTU

- 配置IP MTU
- 验证IP MTU

排除IP MTU故障

- 拓扑
- IP 分段

思科漏洞 ID

相关信息

简介

本文档介绍如何了解Catalyst 9000系列交换机上的最大传输单元(MTU)并对其进行故障排除。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档中的信息基于以下硬件版本：

- C9200

- C9300
- C9400
- C9500
- C9600

注意：您可以使用全局命令system mtu同时配置设备上所有接口的MTU大小。 从Cisco IOS® XE 17.1.1开始，Catalyst 9000交换机支持每端口MTU。每端口MTU支持端口级别和端口通道级别MTU配置。通过每端口MTU，您可以为不同接口以及不同端口通道接口设置不同的MTU值。

注意：有关在其他思科平台上启用这些功能的命令，请参阅相应的配置指南。

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

MTU摘要表

帧总大小= MTU + L2报头

端口类型	默认MTU — 字节	配置的MTU — 字节	L2报头	帧大小总计
L2访问	1500		18	1518
		9216	18	9234
L2中继	1500		22	1522
		9216	22	9238
L3物理端口	1500		18	1518
		9216	18	9234
L3 SVI	1500		18	1518
		9216	18	9234

L3端口上的IP MTU	1500	支持范围。	18	基于ip mtu配置值。

MTU问答

什么是MTU?

- MTU是设备可以转发的最大传输单位。通常，此单位是包括IP报头的IP数据包长度。
- L2报头（如Dot1q标记、MacSec、SVL报头等）不在此计算中。

L2报头及其长度是多少？

- 通用的L2报头是14字节+ 4字节的CRC，总计为18字节。
- TRUNK为dot1q vlan标记增加4个字节，总计为22个字节。
- 同样，MacSec将其自己的报头长度添加到典型的L2报头长度之上。
- SVL端口将其自己的报头长度添加到典型的L2报头长度之上。
- 因此，“Overall Packet on Wire”（线路上的整体数据包）会碰撞到线路。

接口处理的数据包长度是多少？

- Catalyst 9000交换机处理从64字节到9238字节的数据包大小。

什么是默认MTU?

- 默认MTU是交换机在任何用户配置之前设置的MTU。
- 任何Catalyst 9000交换机上的默认MTU为1500字节。
- 以太网端口转发1500字节的第3层数据包+第2层报头。

MTU检查发生在入口还是出口？

出口：MTU是最大传输单位。它是出口检查，决定按原样分段或传输或丢弃出口。

- 如果端口MTU大于要路由的数据包长度，则按原样发送数据包。
- 如果数据包大于出口端口MTU且出口端口为
  - 第3层端口，根据MTU对数据包进行分段。
  - 第2层端口，数据包将被丢弃。（分段仅在第3层完成）

注意：如果数据包在IP报头中设置了Do not Fragment(DNF)位，并且端口MTU小于要路由的数据包，则丢弃数据包

入口：对于到达接口的数据包，也会执行MTU检查。

- 如果接口在其配置的MTU上收到数据包，这些数据包将被视为超大数据包并被丢弃。

什么是巨型数据包？

- 在Catalyst 9000交换机上，任何超过1500字节的数据包都是巨型数据包或巨型数据包。
  - 示例1:如果接口MTU配置为转发大小为9216字节的巨型帧，它将接受或发送9216字节的

帧+第2层报头。
    - 示例2:如果接口MTU配置为转发大小为5000字节的巨型帧，它将接受或发送5000字节的帧+第2层报头。

Jumbo数据包或Oversized数据包是否被视为错误数据包？

- 接口丢弃通过已配置的MTU接收的数据包并将数据包报告为错误。
- 如果接口配置为传输巨型MTU，且收到的数据包在此值内，则不会将其计为错误。

端口可以处理的最小数据包大小是多少？

- 64字节（包括L2报头）是交换机在入口上接受的最小有效数据包大小。
- 如果数据包在线路上包含少于64个字节的数据包，则将其视为残帧，并在入口丢弃。
- 如果数据包应该向外传输，且数据包少于64字节，则交换机会对数据包进行填充，使其在传输之前至少达到64字节。

当系统MTU为9216且SVL报头添加额外的64字节时，会发生什么情况？

- 第3层IP报头下的任何报头都不会计入MTU计算。
- SVL链路可以传输9216 + L2报头+ 64字节的SVL报头。

什么是IP MTU?

- IP MTU仅适用于IP数据包。其他非IP数据包大小不在此命令中。
- 对于IP数据包，IP MTU优先于系统MTU或每端口MTU。
- IP MTU设置IP数据包在需要分段之前可以达到的最大大小。
- 如果物理或逻辑第3层接口的MTU为1500字节，ip mtu为1400字节，则无论系统或每端口MTU设置如何，分段边界都是1400字节。
- MTU是需要与对等路由器/交换机匹配的值。如果对等设备不支持更高的MTU值，请使用IP MTU或MTU匹配两个设备功能。
- 当配置了IP MTU时，设备会将路由协议数据包大小设置为配置的ip MTU值。某些路由协议依靠匹配的MTU值建立路由协议邻居关系。

示例：

- 示例 1：如果接口IP MTU配置为500字节，接口MTU为默认值（无每个端口mtu），系统MTU为9000，则接口MTU为9000字节，IP分段为500字节。
- 示例 2：GRE隧道是出口接口，因此24字节的GRE报头需要计入数据包大小计算（ip mtu 1476 + 24字节的GRE报头= 1500总MTU）。

系统MTU和每端口MTU有何区别？

- 系统MTU是全局配置，用于设置整个设备的MTU。这会将所有前面板物理端口和逻辑端口更改为system mtu命令设置的值。
- 每个端口的MTU允许按每个接口设置MTU值，这优先于系统MTU配置。删除每个端口的设置后，接口将返回到系统MTU。

示例：

- 示例 1：系统MTU值设置为9000，所有物理和逻辑端口MTU设置为9000。
- 示例 2：如果接口的MTU配置为4000，系统MTU配置为9000，则接口将使用MTU 4000，而其他端口将使用MTU 9000。

由于MTU限制，分段会产生什么影响？

- 设备通常在数据平面中转发已分片的数据包，但是，如果设备负责分片或重组，则可能会出现性能/资源问题。
- 分段可能对负责分段处理的应用和设备的整体吞吐量和性能产生严重影响。
- 许多平台中的分段数据包处理在软件中完成，需要花费大量CPU周期来分段或组合分段数据包。
- 如果您的网络经历大量分段，请确保相应地调整MTU以匹配无分段的端到端数据包流。

什么是PMTUD（路径MTU发现）？

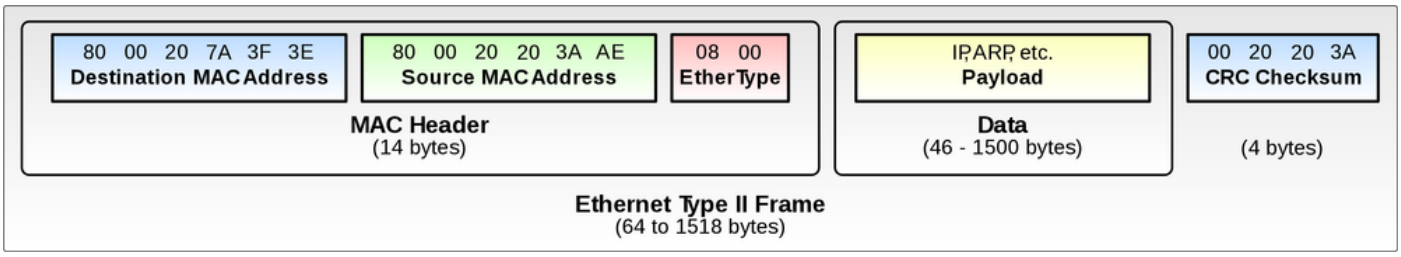
- 如前所述，TCP MSS 负责处理 TCP 连接的两个终端上的分段，但不处理在这两个终端中间有一个较小的 MTU 链路的情况。为了避免在终端之间的路径上出现分段，开发了 PMTUD。它用于动态确定从数据包源到数据包目的地路径上的最小 MTU。
- 有关PMTUD以及如何进行故障排除的详细信息，请参阅解决GRE和IPsec的IPv4分段、MTU、MSS和PMTUD问题。

IPv6 MTU

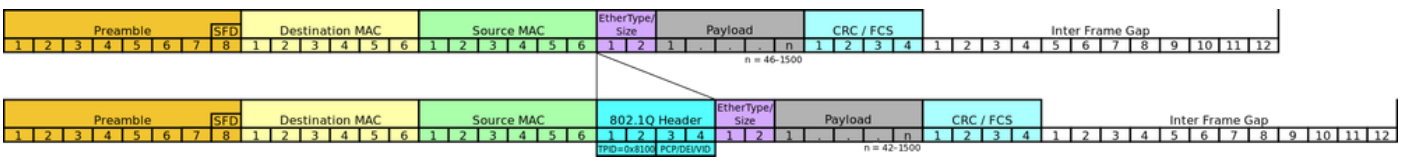
- IPv6 MTU的运行方式与IP MTU相同。
- 要配置，请在接口配置下使用ipv6 mtu而不是ip mtu。
- IPv6 MTU的最小大小为1280，而IPv4为832字节。
- IPv6 PMTUD的工作方式类似于IPv4。有关详细信息，请参阅IP路由配置指南，Cisco IOS® XE Amsterdam 17.3.x（Catalyst 9500交换机）

以太网帧

标准以太网帧，不带Dot1Q或其他标记。



Dot1Q以太网帧



配置和验证MTU

配置MTU

此配置可以全局执行，也可以使用Cisco IOS® XE 17.1.1或更高版本在每个端口级别执行。请检查您的硬件是否支持此配置。

- 删除端口特定配置后，端口将使用全局系统MTU设置。

```
<#root>
### Global System MTU set to 1800 bytes ###


9500H(config)#
system mtu ?

 <1500-9216> MTU size in bytes
   <-- Size range that is configurable


9500H(config)#
system mtu 1800     <-- Set global to 1800 bytes


Global Ethernet MTU is set to 1800 bytes

.
Note: this is the Ethernet payload size, not the total
Ethernet frame size, which includes the Ethernet
header/trailer and possibly other tags, such as ISL or
802.1q tags.
<-- CLI provides information about what is counted as MTU


### Per-Port MTU set to 9216 bytes ###
9500H(config)#
int TwentyFiveGigE1/0/1

9500H(config-if)#
mtu 9126        <-- Interface specific MTU configuration
```

验证MTU

本节介绍如何验证MTU的软件和硬件设置。

- 验证软件配置的MTU和硬件MTU。
- 如果硬件与软件中配置的MTU不匹配，可能会发生流量丢失。

软件MTU验证

<#root>

9500H#show system mtu

Global Ethernet MTU is

1800 bytes

.

<-- Global level MTU

9500H#

show interfaces mtu

Port Name MTU
Twe1/0/1

9216 <-- Per-Port MTU override

Twe1/0/2

1800 <-- No per-port MTU uses global MTU

<...snip...>

9500H#

show interfaces TwentyFiveGigE 1/0/1 | inc MTU
MTU 9216

bytes, BW 1000000 Kbit/sec, DLY 10 usec,

9500H#

show interfaces TwentyFiveGigE 1/0/2 | inc MTU
MTU 1800 bytes,

BW 25000000 Kbit/sec, DLY 10 usec,

硬件MTU验证

```
<#root>

9500H#

show platform software fed active ifm mappings


Interface

IF_ID

    Inst Asic Core Port SubPort Mac Cntx LPN GPN Type Active
TwentyFiveGigE1/0/1

 0x8

        1   0   1    20     0      16  4    1   101 NIF  Y

<-- Retrieve the IF_ID for use in the next command


TwentyFiveGigE1/0/2

0x9

        1   0   1    21     0      17  5    2   102 NIF  Y

9500H#

show platform software fed active ifm if-id 0x8 | inc MTU


                Jumbo MTU .............

[9216]  <-- Hardware matches software configuration


9500H#

show platform software fed active ifm if-id 0x9 | in MTU

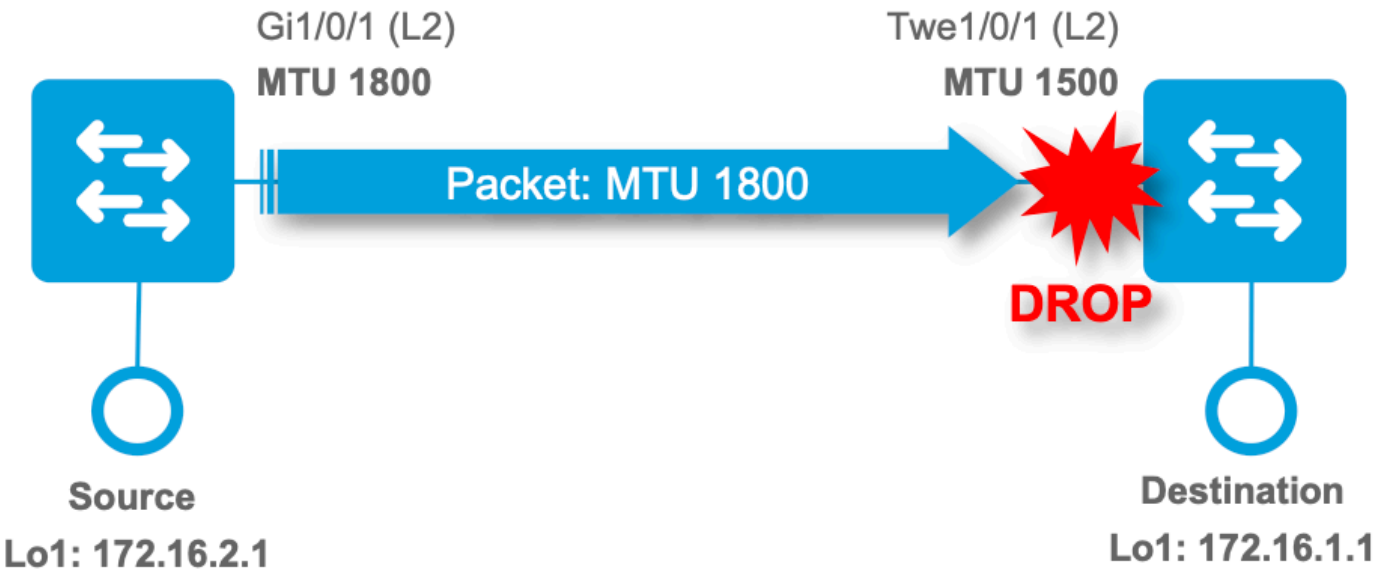
                Jumbo MTU .............

[1800]  <-- Hardware matches software configuration
```

注意：Show platform software fed <active|standby>可能不相同。某些平台需要show platform hardware fed switch <active|standby|sw_num>。

排除MTU故障

拓扑



入口数据包丢弃（入口MTU更低）

如果其中任一计数器增加，通常意味着收到的数据包已到达配置的MTU。

- show interface命令中的Giants计数器。
- show controller命令中的ValidOverSize计数器。

<#root>

```
9500H#

show int twentyFiveGigE 1/0/3 | i MTU
MTU 1500 bytes,

BW 100000 Kbit/sec, DLY 100 usec,
   0 runts,

0 giants

, 0 throttles

<-- No giants counted


9500H#

show controllers ethernet-controller twentyFiveGigE 1/0/3 | i ValidOverSize

            0 Deferred frames

0 ValidOverSize frames <-- No giants counted


### 5 pings from neighbor device with MTU 1800 to ingress port MTU 1500 ###


9500H#
```

```
show int twentyFiveGigE 1/0/3 | i MTU|giant

MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
   0 runts,

5 giants

, 0 throttles

<-- 5 giants counted



9500H#

show controllers ethernet-controller twentyFiveGigE 1/0/3 | i ValidOverSize


            0 Deferred frames

5 ValidOverSize frames <-- 5 giants counted
```

有关show controllers ethernet-controller命令的详细信息。

- 如果数据包通过已配置的MTU到达，并且未通过CRC检查，则将其计为InvalidOverSize。
- 如果数据包到达已配置的MTU中并且未通过CRC检查，则将其计为FcsErr。

```
<#root>

9500H#

show controllers ethernet-controller twentyFiveGigE 1/0/3 | i Fcs|InvalidOver


            0 Good (>1 coll) frames

0 InvalidOverSize frames <-- MTU too large and bad CRC


            0 Gold frames dropped

 0 FcsErr frames          <-- MTU within limits with bad CRC
```

配置和验证IP MTU

配置IP MTU

本节介绍如何在隧道接口上配置ip MTU。

- IP MTU可配置为影响本地系统生成的IP数据包的大小（例如路由协议更新），或者可用于设置分段时的大小。

```
<#root>
```

```
C9300(config)#
interface tunnel 1

C9300(config-if)#
ip mtu 1400



interface Tunnel1

 ip address 10.11.11.2 255.255.255.252

ip mtu 1400                        <-- IP MTU command sets this line at 1400

 ip ospf 1 area 0
 tunnel source Loopback0
 tunnel destination 192.168.1.1
```

验证IP MTU

软件IP MTU验证

<#root>

```
C9300#
sh ip interface tunnel 1 <-- Show the IP level configuration of the interface

Tunnel1 is up, line protocol is up
  Internet address is 10.11.11.2/30
  Broadcast address is 255.255.255.255
  Address determined by setup command

  MTU is 1400 bytes <-- max size of IP packet before fragmentation occurs
```

硬件IP MTU验证

<#root>

```
C9300#sh platform software fed switch active ifm interfaces tunnel
Interface
IF_ID
              State
------------------------------------------------------------------------
Tunnel1
0x00000050
```

```
          READY
<-- Retrieve the IF_ID for use in the next command

C9300#sh platform software fed switch active ifm if-id 0x00000050
Interface IF_ID

         : 0x0000000000000050

<-- The interface ID (IF_ID)


Interface Name          : Tunnel1


Interface Block Pointer : 0x7fe98cc2d118
Interface Block State   : READY
Interface State         : Enabled
Interface Status        : ADD, UPD
Interface Ref-Cnt       : 4

Interface Type          : TUNNEL

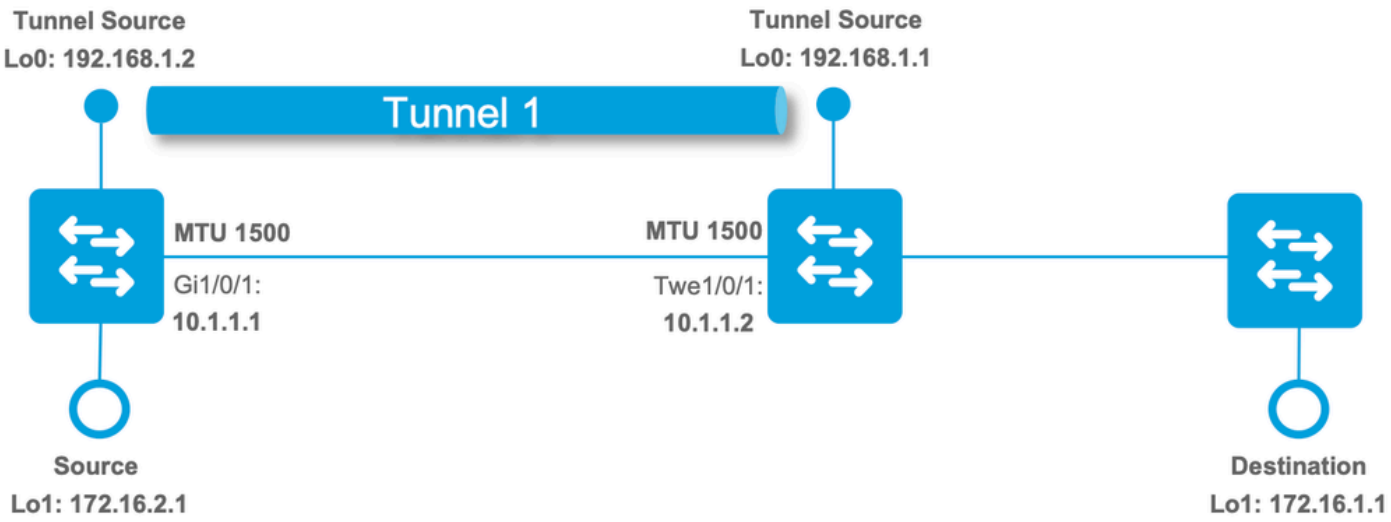

<...snip...>


       Tunnel Sub-mode: 0 [none]
       Hw Support : Yes
       Tunnel Vrf : 0


 IPv4 MTU : 1400                        <-- Hardware matches software configuration
<...snip...>
```

排除IP MTU故障

拓扑



IP 分段

当通过隧道接口发送数据包时，分段可能会以以下示例中提到的两种方式发生。

标准IP分段

将原始数据包分段，以便在隧道封装之前减少MTU。

- 只有入口设备负责此分段操作，分段将在实际终端而不是隧道终端重组。
- 这种数据包分段不需要如此多的资源。

<#root>

Tunnel Source Device: Tunnel IP MTU 1400 | Interface MTU 1500

C9300#

ping 172.16.1.1 source Loopback 1 size 1500 repeat 10 <-- ping with size over IP MTU 1400

Type escape sequence to abort.
Sending 100, 1500-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.2.1
!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/1 ms

Tunnel Destination Device: Ingress Capture Twe1/0/1

9500H#

show monitor capture 1

Status Information for Capture 1

Target Type:

Interface: TwentyFiveGigE1/0/1, Direction: IN <-- Ingress Physical interface

9500H#sh monitor capture 1 buffer br | inc IPv4|ICMP

9 22.285433 172.16.2.1 b^F^R 172.16.1.1

IPv4 1434 Fragmented IP protocol (proto=ICMP 1, off=0, ID=6c03)

10 22.285526 172.16.2.1 b^F^R 172.16.1.1 ICMP 162 Echo (ping) request id=0x0004, seq=0/0, ttl=255

11 22.286295 172.16.2.1 b^F^R 172.16.1.1

IPv4 1434 Fragmented IP protocol (proto=ICMP 1, off=0, ID=6c04)

12 22.286378 172.16.2.1 b^F^R 172.16.1.1 ICMP 162 Echo (ping) request id=0x0004, seq=1/256, ttl=2

<-- Fragmentation occurs on the Inner ICMP packet

(proto=ICMP 1)

<-- Fragments are not reassembled until they reach the actual endpoint device 172.16.1.1

Post隧道封装分段

发生封装后，实际隧道数据包分段以减少MTU，但设备检测到MTU过大。

- 在这种情况下，隧道目标是负责分段重组的设备，而不是真正的目标终端。
- 出现配置问题时会发生这种情况。在应用隧道报头后，设备设置的IP MTU高于实际端口或系统MTU可以处理的IP MTU。
- 在这种情况下，隧道源必须对隧道本身进行分段，隧道目标必须重组隧道报头以便将数据包发送到下一跳或目标。
- 这种报头分段会增加大量的处理开销；这取决于必须处理的流的速率。
- 根据平台、代码和流量速率，您还可以看到CoPP Class Forus流量中的丢包和丢包。

<#root>

Tunnel Source Device: Tunnel IP MTU 1500 | Interface MTU 1500

C9300(config-if)#

ip mtu 1500

%Warning: IP MTU value set 1500 is greater than the current transport value 1476, fragmentation may occu
<-- Device warns the user that this can cause fragmentation (this is a configuration issue)

Tunnel Destination Device: Ingress Capture Twe1/0/1

9500H#

show monitor capture 1

Status Information for Capture 1
Target Type:

Interface: TwentyFiveGigE1/0/1, Direction: IN <-- Ingress Physical interface

9500H

#sh monitor capture 1 buffer br | i IPv4|ICMP

    1   0.000000

192.168.1.2 b^F^R 192.168.1.1

IPv4 1514 Fragmented IP protocol (proto=Generic Routing Encapsulation 47

, off=0, ID=4501)
    2   0.000042   172.16.2.1 b^F^R 172.16.1.1   ICMP 60 Echo (ping) request id=0x0005, seq=0/0, ttl=25
    3   2.000598

192.168.1.2 b^F^R 192.168.1.1

IPv4 1514 Fragmented IP protocol (proto=Generic Routing Encapsulation 47

, off=0, ID=4502)
    4   2.000642   172.16.2.1 b^F^R 172.16.1.1   ICMP 60 Echo (ping) request id=0x0005, seq=1/256, ttl=

<-- Fragmentation has occurred on the outer GRE header(proto=Generic Routing Encapsulation 47)
<-- Fragments must be reassembled at the Tunnel endpoint, in this case the 9500

思科漏洞 ID

重新加载后不考虑Cisco Bug ID CSCvr84911 System MTU。

思科漏洞ID CSCvq30464CAT9400:MTU配置未应用于变为活动状态的非活动端口。

Cisco Bug ID CSCvh04282 Cat9300非默认系统MTU配置值在重新加载后不受影响。

相关信息

- 技术支持和文档 - Cisco Systems
- 接口和硬件组件配置指南，Cisco IOS® XE Amsterdam 17.3.x（Catalyst 9500交换机）
- 接口和硬件组件配置指南，Cisco IOS® XE Amsterdam 17.3.x（Catalyst 9600交换机）
- 解决 GRE 和 IPsec 中的 IPv4 分段、MTU、MSS 和 PMTUD 问题

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）。