对Catalyst 9K交换机中MSS调整引起的TCP慢度问题进行故障排除

目录

简介

本文档介绍Catalyst 9K交换机如何执行TCP MSS调整，以及TCP慢度如何与此功能相关联。

有关TCP MSS调整的信息

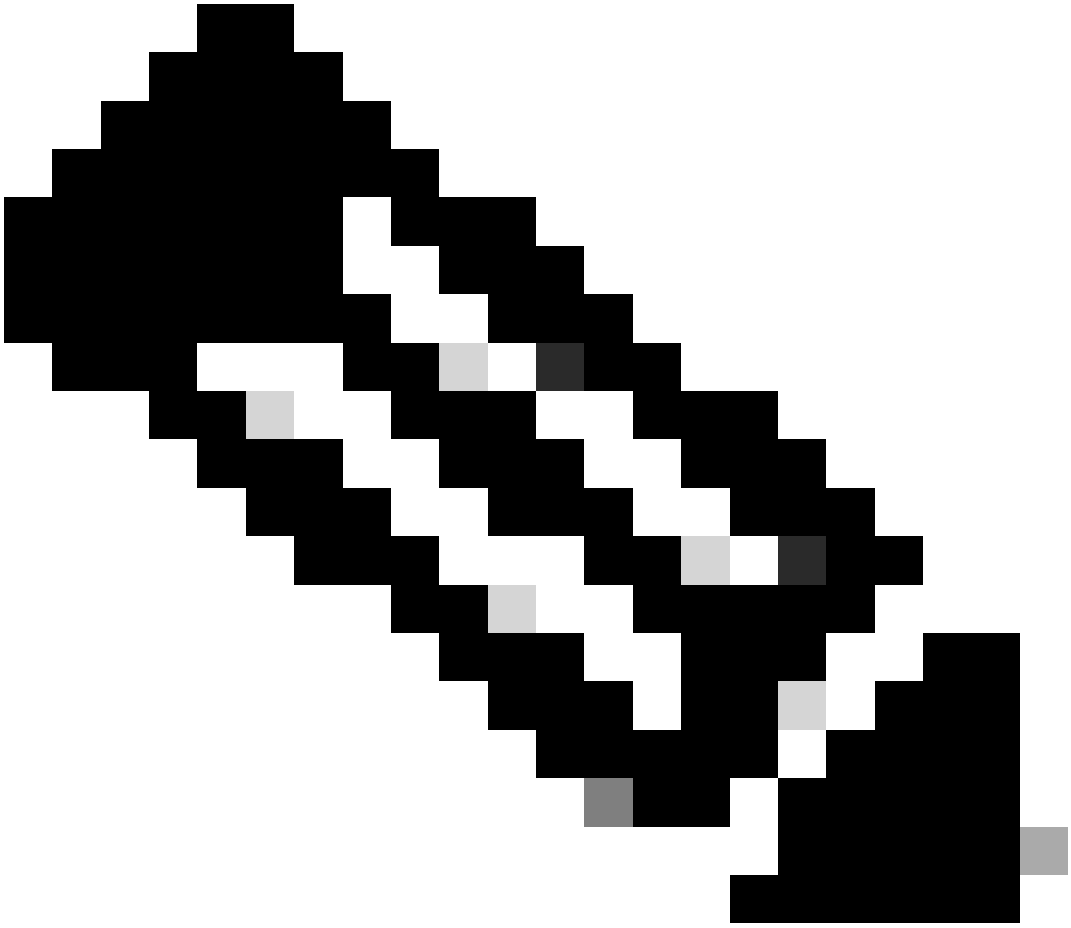
通过传输控制协议(TCP)最大分段大小(MSS)调整功能，可以配置通过路由器的临时数据包的最大分段大小，尤其是设置了SYN位的TCP分段。ip tcp adjust-mss命令在接口配置模式下使用，用于指定SYN数据包的中间路由器上的MSS值，以避免截断。

当主机（通常是PC）与服务器发起TCP会话时，它会使用TCP SYN数据包中的MSS选项字段协商IP分段大小。主机上的MTU配置决定了MSS字段的值。PC网卡的默认MTU值为1500字节，TCP MSS值为1460（1500字节- 20字节IP报头- 20字节TCP报头）。

以太网PPP (PPPoE)标准仅支持1492字节的MTU。

主机和PPPoE MTU大小之间的差异会导致主机和服务器之间的路由器丢弃1500字节的数据包并终止通过PPPoE网络的TCP会话。
即使主机上启用了路径MTU（可检测路径中的正确MTU），会话也可以被丢弃，因为系统管理员有时会禁用必须从主机中继的互联网控制消息协议(ICMP)错误消息才能使路径MTU正常工作。

ip tcp adjust-mss命令通过调整TCP SYN数据包的MSS值来帮助防止TCP会话被丢弃。ip tcp adjust-mss命令仅对通过路由器的TCP连接有效。在大多数情况下，ip tcp adjust-mss命令的max-segment-size参数的最佳值为1452字节。
此值加上20字节的IP报头、20字节的TCP报头和8字节的PPPoE报头，构成一个1500字节的数据包，与以太网链路的MTU大小匹配。



注意：基于TCP MSS调整的流量在Catalyst 9K交换机中进行软件交换。本文档说明假设基于TCP MSS调整的流量采用软件交换的场景。请参阅配置指南，以确认特定硬件/软件是否切换基于TCP MSS调整的流量。

行为

如前所述，基于TCP MSS调整的流量始终采用软件交换。
这意味着如果您尝试并执行TCP调整，则交换机将TCP流量发送到CPU以进行MSS修改。

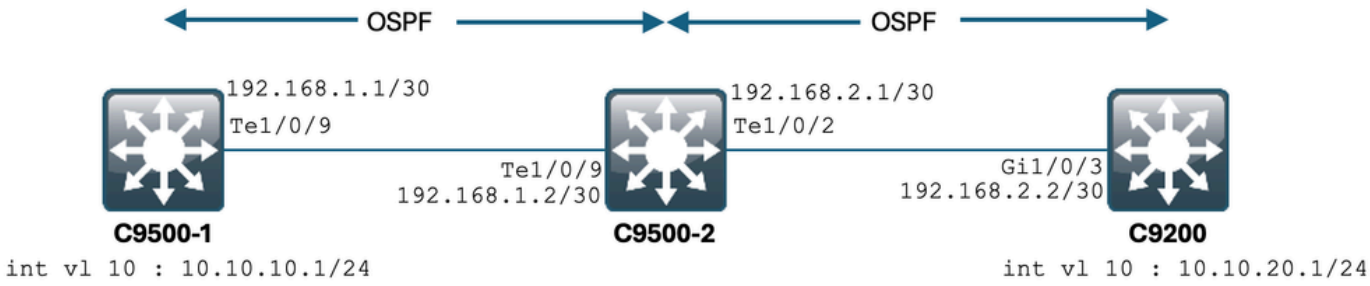
例如，如果修改接口上的TCP MSS值，则该接口上收到的所有TCP流量都会被传送到CPU。
然后，CPU更改MSS值，并将流量发送到该TCP数据包发往的所需接口。

因此，如果MSS调整存在大量的TCP流量，则会重载CPU队列。
当CPU队列过载时，控制平面监察器(COPP)会管制流量并丢弃数据包，以保持队列监察器速率。
这会导致TCP数据包被丢弃。

因此，可以看到文件传输缓慢、SSH会话创建和Citrix应用缓慢（如果使用TCP）等问题。

这里显示的是如何实现此操作的真实示例。

拓扑



场景

您将从C9500-1通过SSH进入C9200。
使用C9500-1的VLAN 10 (10.10.10.1)作为源的SSH。
SSH的目的地是C9200的VLAN 20 (10.10.20.1/24)。
SSH是基于TCP的，因此TCP中的任何速度慢也会影响此SSH会话的创建。

C9500-1和C9200之间有一个中转L3交换机(C9500-2)。
有两个中转/30第3层链路，一个在C9500-1和C9500-2之间，另一个在C9500-2和C9200之间。
OSPF用于实现所有三台交换机之间的可达性，并且所有/30个子网和SVI都通告到OSPF中。
前面显示的所有IP在它们之间均可访问。

在C9500-2 Te1/0/9中，对TCP MSS值进行了修改。
从C9500-1发起SSH时，会发生TCP三次握手。
SYN数据包到达C9500-2 Te1/0/9（入口），在该处执行TCP MSS调整。

初始配置和行为

对C9500-2 Te1/0/9（双向）执行EPC捕获，并启动了从C9500-1到C9200的SSH。

以下是EPC配置：

```
C9500-2#show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: TenGigabitEthernet1/0/9, Direction: BOTH
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 80
```

```
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
C9500-2#
```

启动EPC：

```
C9500-2#monitor capture mycap start
Started capture point : mycap
C9500-2#
```

启动从C9500-1到C9200的SSH：

```
C9500-1#ssh -l admin 10.10.20.1
Password:
```

停止EPC：

```
C9500-2#monitor capture mycap stop
Capture statistics collected at software:
Capture duration - 6 seconds
Packets received - 47
Packets dropped - 0
Packets oversized - 0
Bytes dropped in asic - 0
Capture buffer will exists till exported or cleared
Stopped capture point : mycap
C9500-2#
```

下面是EPC捕获的数据包：

```
C9500-2#show monitor capture mycap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000 10.10.10.1 -> 10.10.20.1 TCP 60 44274 -> 22 [SYN] Seq=0 Win=4128 Len=0 MSS=536
2 0.001307 10.10.20.1 -> 10.10.10.1 TCP 60 22 -> 44274 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
3 0.001564 10.10.10.1 -> 10.10.20.1 TCP 60 44274 -> 22 [ACK] Seq=1 Ack=1 Win=4128 Len=0
4 0.003099 10.10.20.1 -> 10.10.10.1 SSH 73 Server: Protocol (SSH-2.0-Cisco-1.25)
5 0.003341 10.10.10.1 -> 10.10.20.1 SSH 73 Client: Protocol (SSH-2.0-Cisco-1.25)
6 0.003419 10.10.10.1 -> 10.10.20.1 TCP 118 [TCP segment of a reassembled PDU]
```

```
7 0.003465 10.10.10.1 -> 10.10.20.1 TCP 118 44274 -> 22 [ACK] Seq=84 Ack=20 Win=4109 Len=64 [TCP segmen
8 0.003482 10.10.10.1 -> 10.10.20.1 TCP 118 44274 -> 22 [ACK] Seq=148 Ack=20 Win=4109 Len=64 [TCP segme
9 0.003496 10.10.10.1 -> 10.10.20.1 TCP 118 44274 -> 22 [ACK] Seq=212 Ack=20 Win=4109 Len=64 [TCP segme
10 0.003510 10.10.10.1 -> 10.10.20.1 TCP 118 44274 -> 22 [ACK] Seq=276 Ack=20 Win=4109 Len=64 [TCP segm
11 0.003525 10.10.10.1 -> 10.10.20.1 TCP 118 44274 -> 22 [ACK] Seq=340 Ack=20 Win=4109 Len=64 [TCP segm
12 0.004719 10.10.20.1 -> 10.10.10.1 TCP 60 22 -> 44274 [ACK] Seq=20 Ack=84 Win=4045 Len=0
~ Output Cut ~
```

您可以看到第1、2、3号数据包中发生TCP握手。
数据包1是SYN数据包。
您可以看到它附带的MSS值为536。
还可看到来自C9200的SYN、ACK数据包（数据包编号2），MSS值为536。
此处，MSS值保持不变，且交换机未更改。

TCP MSS调整后的行为

以下是C9500-2 Te1/0/9上的TCP MSS调整配置：

```
C9500-2#sh run int te1/0/9
Building configuration...
Current configuration : 119 bytes
!
interface TenGigabitEthernet1/0/9
no switchport
ip address 192.168.1.2 255.255.255.252
ip tcp adjust-mss 512 ------------------> Here we are changing the MSS value to 512.
```

现在，对C9500-2 Te1/0/9（两个方向）执行EPC捕获，然后从C9500-1到C9200启动SSH。

以下是EPC配置：

```
C9500-2#show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: TenGigabitEthernet1/0/9, Direction: BOTH
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 80
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
C9500-2#
```

开始捕获，从C9500-1通过SSH连接到C9200，然后停止捕获。

下面是CPU捕获的数据包：

```
C9500-2#show monitor capture mycap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000 b8:a3:77:ec:ba:f7 -> 01:00:0c:cc:cc:cc CDP 398 Device ID: C9500-1.cisco.com Port ID: TenGiga
2 0.636138 10.10.10.1 -> 10.10.20.1 TCP 60 53865 -> 22 [SYN] Seq=0 Win=4128 Len=0 MSS=536
3 0.637980 10.10.20.1 -> 10.10.10.1 TCP 60 22 -> 53865 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=512
4 0.638214 10.10.10.1 -> 10.10.20.1 TCP 60 53865 -> 22 [ACK] Seq=1 Ack=1 Win=4128 Len=0
5 0.639997 10.10.20.1 -> 10.10.10.1 SSH 73 Server: Protocol (SSH-2.0-Cisco-1.25)
6 0.640208 10.10.10.1 -> 10.10.20.1 SSH 73 Client: Protocol (SSH-2.0-Cisco-1.25)
7 0.640286 10.10.10.1 -> 10.10.20.1 TCP 118 [TCP segment of a reassembled PDU]
8 0.640341 10.10.10.1 -> 10.10.20.1 TCP 118 53865 -> 22 [ACK] Seq=84 Ack=20 Win=4109 Len=64 [TCP segmen
9 0.640360 10.10.10.1 -> 10.10.20.1 TCP 118 53865 -> 22 [ACK] Seq=148 Ack=20 Win=4109 Len=64 [TCP segme
10 0.640375 10.10.10.1 -> 10.10.20.1 TCP 118 53865 -> 22 [ACK] Seq=212 Ack=20 Win=4109 Len=64 [TCP segm
11 0.640390 10.10.10.1 -> 10.10.20.1 TCP 118 53865 -> 22 [ACK] Seq=276 Ack=20 Win=4109 Len=64 [TCP segm
12 0.640410 10.10.10.1 -> 10.10.20.1 TCP 118 53865 -> 22 [ACK] Seq=340 Ack=20 Win=4109 Len=64 [TCP segm
~ Output Cut ~
```

您可以看到2、3、4号数据包中发生了TCP握手。
数据包2是SYN数据包。
您可以看到它附带的MSS值为536。
但是，可以看到SYN、ACK数据包（数据包编号3）来自C9200，MSS值为512。
这是因为，当SYN数据包到达C9500-2 Te1/0/9时，会将其发送到C9500-2的CPU，以将TCP MSS从536修改为512。
C9500-2的CPU将MSS更改为512，并将SYN数据包从Te1/0/2发送到C9200。
然后，以下所有TCP事务使用相同的修改MSS值。

现在，让我们深入探讨SYN数据包如何通过交换机并发生MSS更改。
一旦此SYN数据包到达C9500-2的接口，它将被发送到CPU以进行MSS修改。
它首先通过FED（您可以捕获的位置），然后进入CPU（也可以捕获的位置）。

我们首先来看看在C9500-2上捕获FED弃踢。

以下是FED传送捕获配置：

```
C9500-2#debug platform software fed switch 1 punt packet-capture buffer limit 16384
Punt PCAP buffer configure: one-time with buffer size 16384...done
```

启动FED传送捕获：

```
C9500-2#debug platform software fed switch 1 punt packet-capture start
Punt packet capturing started.
```

启动从C9500-1到C9200的SSH：

```
C9500-1#ssh -l admin 10.10.20.1
Password:
```

停止美联储弃权捕获：

```
C9500-2#debug platform software fed switch 1 punt packet-capture stop
Punt packet capturing stopped. Captured 3 packet(s)
```

下面是FED传送捕获的数据包：

```
C9500-2#show platform software fed switch active punt packet-capture brief
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 3 packets. Capture capacity : 16384 packets

------ Punt Packet Number: 1, Timestamp: 2024/07/31 01:29:46.373 ------
interface : physical: TenGigabitEthernet1/0/9[if-id: 0x00000040], pal: TenGigabitEthernet1/0/9 [if-id:
metadata : cause: 55 [For-us control], sub-cause: 0, q-no: 4, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: 0100.5e00.0005, src mac: b8a3.77ec.baf7
ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 224.0.0.5, src ip: 192.168.1.1
ipv4 hdr : packet len: 100, ttl: 1, protocol: 89

------ Punt Packet Number: 2, Timestamp: 2024/07/31 01:29:47.432 ------
interface : physical: TenGigabitEthernet1/0/9[if-id: 0x00000040], pal: TenGigabitEthernet1/0/9 [if-id:
metadata : cause: 11 [For-us data], sub-cause: 1, q-no: 14, linktype: MCP_LINK_TYPE_IP [1]
ether hdr : dest mac: 00a3.d144.4bf7, src mac: b8a3.77ec.baf7
ether hdr : ethertype: 0x0800 (IPv4)
ipv4 hdr : dest ip: 10.10.20.1, src ip: 10.10.10.1
ipv4 hdr : packet len: 44, ttl: 254, protocol: 6 (TCP)
tcp hdr : dest port: 22, src port: 35916

------ Punt Packet Number: 3, Timestamp: 2024/07/31 01:29:48.143 ------
interface : physical: TenGigabitEthernet1/0/1[if-id: 0x00000009], pal: TenGigabitEthernet1/0/1 [if-id:
metadata : cause: 96 [Layer2 control protocols], sub-cause: 0, q-no: 1, linktype: MCP_LINK_TYPE_LAYER2
ether hdr : dest mac: 0100.0ccc.cccc, src mac: 78bc.1a27.c203
ether hdr : length: 443
```

您可以看到数据包2是来自Te1/0/9的从10.10.10.1到10.10.20.1的TCP SYN数据包。
此处必须注意“q-no”。您可以看到，它选择队列14从FED进入CPU。

在这里，您可以看到存在流量从FED移动到CPU的所有32个队列：

```
C9500-2#show platform hardware fed switch active qos queue stats internal cpu policer

CPU Queue Statistics
============================================================================================
(default) (set) Queue Queue
QId PlcIdx Queue Name Enabled Rate Rate Drop(Bytes) Drop(Frames)
--------------------------------------------------------------------------------------------
0 11 DOT1X Auth Yes 1000 1000 0 0
1 1 L2 Control Yes 2000 2000 0 0
2 14 Forus traffic Yes 4000 4000 0 0
3 0 ICMP GEN Yes 600 600 0 0
4 2 Routing Control Yes 5400 5400 0 0
5 14 Forus Address resolution Yes 4000 4000 0 0
6 0 ICMP Redirect Yes 600 600 0 0
7 16 Inter FED Traffic Yes 2000 2000 0 0
8 4 L2 LVX Cont Pack Yes 1000 1000 0 0
9 19 EWLC Control Yes 13000 13000 0 0
10 16 EWLC Data Yes 2000 2000 0 0
11 13 L2 LVX Data Pack Yes 1000 1000 0 0
12 0 BROADCAST Yes 600 600 0 0
13 10 Openflow Yes 200 200 0 0
14 13 Sw forwarding Yes 1000 1000 0 0
15 8 Topology Control Yes 13000 13000 0 0
16 12 Proto Snooping Yes 2000 2000 0 0
17 6 DHCP Snooping Yes 400 400 0 0
18 13 Transit Traffic Yes 1000 1000 0 0
19 10 RPF Failed Yes 200 200 0 0
20 15 MCAST END STATION Yes 2000 2000 0 0
21 13 LOGGING Yes 1000 1000 0 0
22 7 Punt Webauth Yes 1000 1000 0 0
23 18 High Rate App Yes 13000 13000 0 0
24 10 Exception Yes 200 200 0 0
25 3 System Critical Yes 1000 1000 0 0
26 10 NFL SAMPLED DATA Yes 200 200 0 0
27 2 Low Latency Yes 5400 5400 0 0
28 10 EGR Exception Yes 200 200 0 0
29 5 Stackwise Virtual OOB Yes 8000 8000 0 0
30 9 MCAST Data Yes 400 400 0 0
31 3 Gold Pkt Yes 1000 1000 0 0
```

您可以看到开销，队列14是“Sw forwarding”队列。
在这种情况下，TCP流量使用此队列来传送到CPU。

现在，让我们对C9500-2执行CPU（控制平面）捕获。

CPU捕获配置如下：

```
C9500-2#sh mon cap test
Status Information for Capture test
Target Type:
Interface: Control Plane, Direction: BOTH
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 80
```

```
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Packet sampling rate: 0 (no sampling)
C9500-2#
```

开始捕获，从C9500-1通过SSH连接到C9200，然后停止捕获。

下面是CPU捕获的数据包：

```
C9500-2#show monitor capture test buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit
1 0.000000 00:a3:d1:44:4b:81 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost = 0
2 0.000010 00:a3:d1:44:4b:a3 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost = 0
3 0.000013 00:a3:d1:44:4b:a4 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost = 0
4 0.000016 00:a3:d1:44:4b:a6 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost = 0
5 0.000019 00:a3:d1:44:4b:a7 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost = 0
6 0.000022 00:a3:d1:44:4b:a8 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost = 0
7 0.055470 c0:8b:2a:04:f0:6c -> 01:80:c2:00:00:0e LLDP 117 TTL = 120 SysName = bgl18-cx-amx-b02-2.cisco
9 0.220331 28:63:29:20:31:39 -> 00:01:22:53:74:20 0x3836 30 Ethernet II
10 0.327316 192.168.1.1 -> 224.0.0.5 OSPF 114 Hello Packet
11 0.442986 c0:8b:2a:04:f0:68 -> 01:80:c2:00:00:0e LLDP 117 TTL = 120 SysName = bgl18-cx-amx-b02-2.cisc
12 1.714121 10.10.10.1 -> 10.10.20.1 TCP 60 23098 -> 22 [SYN] Seq=0 Win=4128 Len=0 MSS=536
13 1.714375 10.10.10.1 -> 10.10.20.1 TCP 60 [TCP Out-Of-Order] 23098 -> 22 [SYN] Seq=0 Win=4128 Len=0 M
14 2.000302 00:a3:d1:44:4b:81 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost =
15 2.000310 00:a3:d1:44:4b:a3 -> 01:80:c2:00:00:00 STP 60 RST. Root = 32768/1/00:a3:d1:44:4b:80 Cost =
~ Output Cut ~
```

数据包12是进入CPU（传送）的TCP SYN数据包，默认MSS值为536。
数据包13是CPU在将MSS值修改为512之后发出的TCP SYN数据包（注入）。

您也可以快速执行CPU调试，以便查看此更改的发生情况。

CPU调试配置如下：

```
C9500-2#debug ip tcp adjust-mss
TCP Adjust Mss debugging is on
```

启动从C9500-1到C9200的SSH：

```
C9500-1#ssh -l admin 10.10.20.1
Password:
```

停止CPU调试：

```
C9500-2#undebug all
All possible debugging has been turned off
```

查看调试日志：

```
C9500-2#show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml disabl
No Active Message Discriminator.
No Inactive Message Discriminator.
Console logging: disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
filtering disabled
Buffer logging: level debugging, 230 messages logged, xml disabled,
filtering disabled
Exception Logging: size (4096 bytes)
Count and timestamp logging messages: disabled
File logging: disabled
Persistent logging: disabled
No active filter modules.
Trap logging: level informational, 210 message lines logged
Logging Source-Interface: VRF Name:
TLS Profiles:
Log Buffer (102400 bytes):
*Jul 31 01:46:32.052: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:32.893: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:36.136: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:41.318: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:42.412: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:43.254: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:43.638: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:45.783: TCPADJMSS: Input (process)
*Jul 31 01:46:45.783: TCPADJMSS: orig_mss = 536 adj mss = 512 src_ip = 10.10.10.1 dest_ip = 10.10.20.1
*Jul 31 01:46:45.783: TCPADJMSS: paktype = 0x7F83C7BCBF78
*Jul 31 01:46:50.456: TCPADJMSS: process_enqueue_feature
*Jul 31 01:46:51.985: TCPADJMSS: process_enqueue_feature
C9500-2#
```

您可以看到将原始MSS值536调整为512的开销。

最后，您可以在C9200 Gi1/0/3上捕获EPC捕获，以确认TCP SYN数据包确实带有MSS 512。

以下是EPC配置：

```
C9200#sh mon cap mycap
Status Information for Capture mycap
Target Type:
Interface: GigabitEthernet1/0/3, Direction: BOTH
```

```
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 80
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Packet sampling rate: 0 (no sampling)
C9200#
```

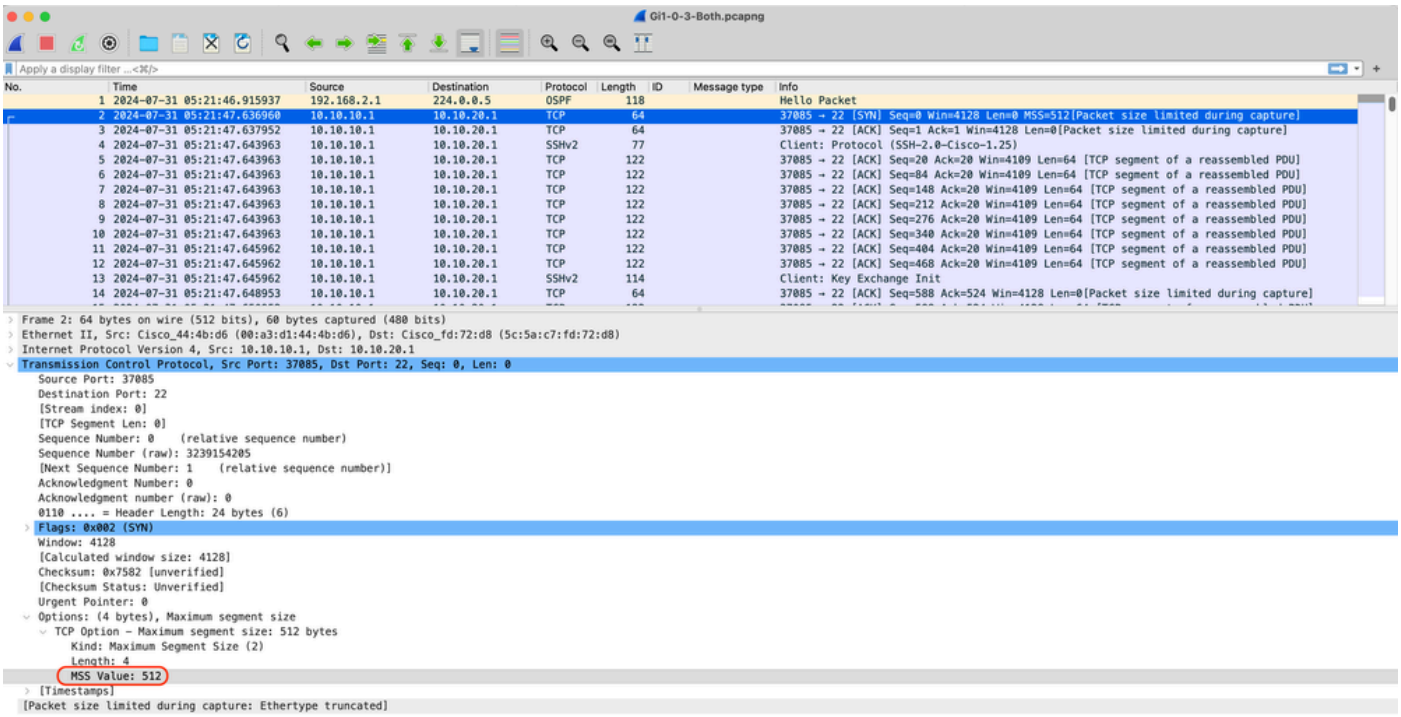
开始捕获，从C9500-1通过SSH连接到C9200，然后停止捕获。

下面是CPU捕获的数据包：

```
C9200#sh mon cap mycap buff br
----------------------------------------------------------------------------
# size timestamp source destination dscp protocol
----------------------------------------------------------------------------
0 118 0.000000 192.168.2.1 -> 224.0.0.5 48 CS6 OSPF
1 64 0.721023 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
2 64 0.722015 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
3 77 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
4 122 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
5 122 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
6 122 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
7 122 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
8 122 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
9 122 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
10 122 0.730025 10.10.10.1 -> 10.10.20.1 48 CS6 TCP
~ Output Cut ~
```

在C9200中，您无法在Wireshark中查看数据包详细信息，只能查看简要和十六进制详细信息。
因此，您可以将早期数据包导出到闪存中的pcap文件。
C9200#mon cap mycap export flash：Gi1-0-3-Both.pcapng
已成功导出

然后，您可以通过TFTP将此文件复制到本地PC，并在Wireshark中打开该文件。
这是Wireshark捕获。



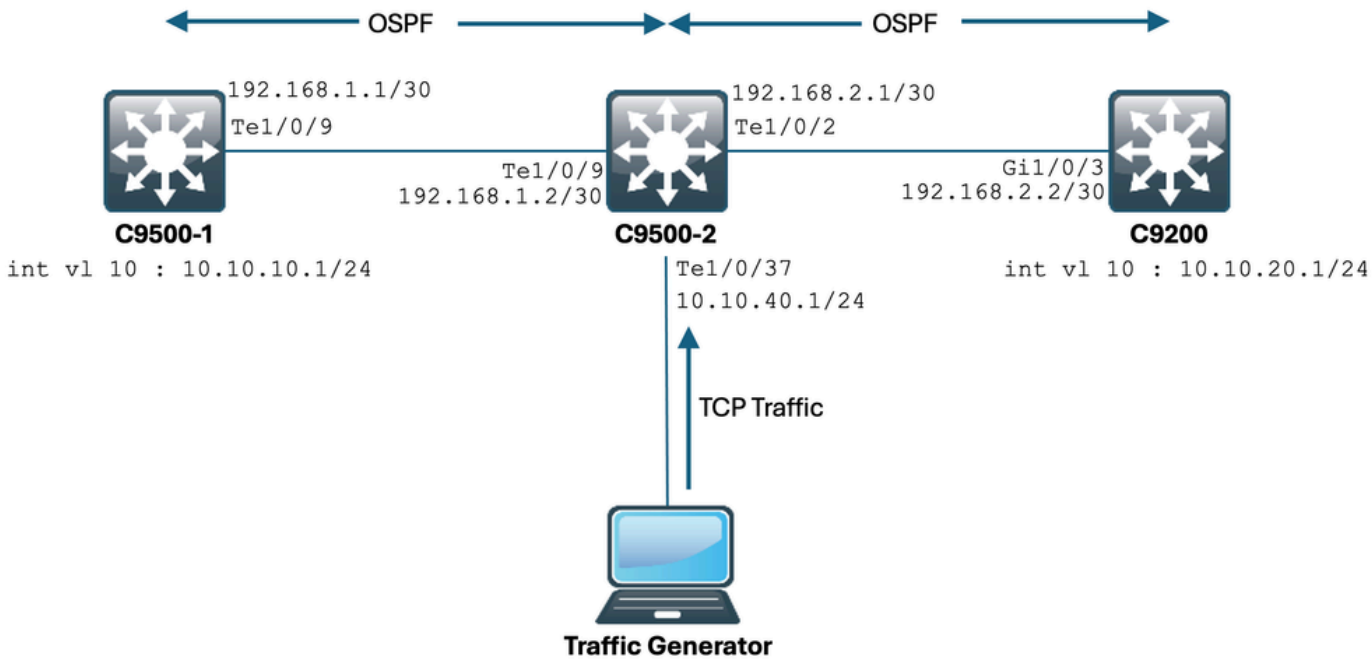
您可以看到SYN数据包的TCP MSS值为512。

TCP MSS调整导致大量TCP流量传输缓慢

现在，让我们假设一个网络有多台使用TCP流量的设备。
例如，它们可以传输文件，或访问基于TCP的应用程序（如Citrix服务器）。

您可以通过将IXIA（流量生成器）连接到C9500-2 Te1/0/37，以高速发送TCP SYN数据包来模拟它。
此IXIA设备用作网段，其中多个用户使用基于TCP的应用。
您已经在Te1/0/37上配置了ip tcp adjust-mss CLI。
这会导致Te1/0/37上接收的所有TCP流量被传送到C9500-2的CPU。
这反过来会阻塞C9500-2的COPP监察器的“软件转发”队列，如本文档前面所述。
因此，从C9500-1到C9200的SSH会话建立会受到影响。
SSH会话未形成且超时，或者在延迟后建立。

拓扑如下所示：



让我们来看看实际操作。

以下是C9500-2 Te1/0/37的配置：

```
C9500-2#sh run int te1/0/37
Building configuration...
Current configuration : 135 bytes
interface TenGigabitEthernet1/0/37
no switchport
ip address 10.10.40.1 255.255.255.0
ip tcp adjust-mss 500
load-interval 30
end
```

现在您开始从IXIA向Te1/0/37接口发送大量流量。
我们来看看传入流量速率：

```
C9500-2#sh int te1/0/37 | in rate
Queueing strategy: fifo
30 second input rate 6425812000 bits/sec, 12550415 packets/sec → We can see the enormous Input rate.
30 second output rate 0 bits/sec, 0 packets/sec
```

现在，让我们尝试从C9500-1通过SSH连接到C9200：

```
C9500-1#ssh -l admin 10.10.20.1
% Connection timed out; remote host not responding
C9500-1#
```

您可以清楚地看到C9500-1无法通过SSH连接到C9200。
这是因为C9500-1发送的TCP SYN数据包被“Sw forwarding”队列丢弃，而该队列正受到来自Te1/0/37的流量的轰炸。

让我们来看看队列：

```
C9500-2#sh platform hardware fed switch active qos queue stats internal cpu policer
CPU Queue Statistics
============================================================================================
(default) (set) Queue Queue
QId PlcIdx Queue Name Enabled Rate Rate Drop(Bytes) Drop(Frames)
--------------------------------------------------------------------------------------------
0 11 DOT1X Auth Yes 1000 1000 0 0
1 1 L2 Control Yes 2000 2000 0 0
2 14 Forus traffic Yes 4000 4000 0 0
3 0 ICMP GEN Yes 600 600 0 0
4 2 Routing Control Yes 5400 5400 0 0
5 14 Forus Address resolution Yes 4000 4000 0 0
6 0 ICMP Redirect Yes 600 600 0 0
7 16 Inter FED Traffic Yes 2000 2000 0 0
8 4 L2 LVX Cont Pack Yes 1000 1000 0 0
9 19 EWLC Control Yes 13000 13000 0 0
10 16 EWLC Data Yes 2000 2000 0 0
11 13 L2 LVX Data Pack Yes 1000 1000 0 0
12 0 BROADCAST Yes 600 600 0 0
13 10 Openflow Yes 200 200 0 0
14 13 Sw forwarding Yes 1000 1000 39683368064 620052629 → We can see the huge number of dropped packets in t
15 8 Topology Control Yes 13000 13000 0 0
16 12 Proto Snooping Yes 2000 2000 0 0
17 6 DHCP Snooping Yes 400 400 0 0
18 13 Transit Traffic Yes 1000 1000 0 0
19 10 RPF Failed Yes 200 200 0 0
20 15 MCAST END STATION Yes 2000 2000 0 0
21 13 LOGGING Yes 1000 1000 0 0
22 7 Punt Webauth Yes 1000 1000 0 0
23 18 High Rate App Yes 13000 13000 0 0
24 10 Exception Yes 200 200 0 0
25 3 System Critical Yes 1000 1000 0 0
26 10 NFL SAMPLED DATA Yes 200 200 0 0
27 2 Low Latency Yes 5400 5400 0 0
28 10 EGR Exception Yes 200 200 0 0
29 5 Stackwise Virtual OOB Yes 8000 8000 0 0
30 9 MCAST Data Yes 400 400 0 0
31 3 Gold Pkt Yes 1000 1000 0 0
```

让我们多次收集输出，以确保丢弃的计数在问题期间增加：

```
C9500-2#sh platform hardware fed switch active qos queue stats internal cpu policer | in Sw forwarding
14 13 Sw forwarding Yes 1000 1000 47046906560 735107915
14 13 21 Sw forwarding Yes
13 system-cpp-police-sw-forward : Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Traffic/
21 system-cpp-police-ios-feature : ICMP GEN/ BROADCAST/ ICMP Redirect/ L2 LVX Cont Pack/ Proto Snooping/
C9500-2#
```

```
!
C9500-2#sh platform hardware fed switch active qos queue stats internal cpu policer | in Sw forwarding
14 13 Sw forwarding Yes 1000 1000 47335535936 739617752
14 13 21 Sw forwarding Yes
13 system-cpp-police-sw-forward : Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Traffic/
21 system-cpp-police-ios-feature : ICMP GEN/ BROADCAST/ ICMP Redirect/ L2 LVX Cont Pack/ Proto Snooping/
C9500-2#
!
C9500-2#sh platform hardware fed switch active qos queue stats internal cpu policer | in Sw forwarding
14 13 Sw forwarding Yes 1000 1000 47666441088 744788145
14 13 21 Sw forwarding Yes
13 system-cpp-police-sw-forward : Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Traffic/
21 system-cpp-police-ios-feature : ICMP GEN/ BROADCAST/ ICMP Redirect/ L2 LVX Cont Pack/ Proto Snooping/
C9500-2#
```

如您所见，丢弃的计数增加，并且SSH流量（TCP SYN数据包）在此处被丢弃。

现在，如果您不知道通过哪个接口/SVI获得这种流量流入，您有一个特定命令可以帮您解决问题。

```
C9500-2#show platform software fed switch active punt rates interfaces
Punt Rate on Interfaces Statistics
Packets per second averaged over 10 seconds, 1 min and 5 mins
================================================================================================
| | Recv | Recv | Recv | Drop | Drop | Drop
Interface Name | IF_ID | 10s | 1min | 5min | 10s | 1min | 5min
================================================================================================
TenGigabitEthernet1/0/37 0x00000042 1000 1000 1000 0 0 0
------------------------------------------------------------------------------------------------
C9500-2#
```

show platform software fed switch active punt rates interfaces命令提供了负责接收发送到CPU的大量流量的接口列表。
您在此处可以清楚地看到Te1/0/37，它是用于获取TCP流量的接口。

现在，如果要查看到达所有COPP监察器队列（在早期接口上接收）的流量量，您可以使用：
show platform software fed switch active punt rates interfaces <IF_ID from the above output>

我们来看看：

```
C9500-2#show platform software fed switch active punt rates interfaces 0x42
Punt Rate on Single Interfaces Statistics
Interface : TenGigabitEthernet1/0/37 [if_id: 0x42]

Received Dropped
-------- -------
Total : 2048742 Total : 0
10 sec average : 1000 10 sec average : 0
1 min average : 1000 1 min average : 0
5 min average : 1000 5 min average : 0

Per CPUQ punt stats on the interface (rate averaged over 10s interval)
```

```
=============================================================================
Q | Queue | Recv | Recv | Drop | Drop |
no | Name | Total | Rate | Total | Rate |
=============================================================================
0 CPU_Q_DOT1X_AUTH 0 0 0 0
1 CPU_Q_L2_CONTROL 7392 0 0 0
2 CPU_Q_FORUS_TRAFFIC 0 0 0 0
3 CPU_Q_ICMP_GEN 0 0 0 0
4 CPU_Q_ROUTING_CONTROL 0 0 0 0
5 CPU_Q_FORUS_ADDR_RESOLUTION 0 0 0 0
6 CPU_Q_ICMP_REDIRECT 0 0 0 0
7 CPU_Q_INTER_FED_TRAFFIC 0 0 0 0
8 CPU_Q_L2LVX_CONTROL_PKT 0 0 0 0
9 CPU_Q_EWLC_CONTROL 0 0 0 0
10 CPU_Q_EWLC_DATA 0 0 0 0
11 CPU_Q_L2LVX_DATA_PKT 0 0 0 0
12 CPU_Q_BROADCAST 0 0 0 0
13 CPU_Q_CONTROLLER_PUNT 0 0 0 0
14 CPU_Q_SW_FORWARDING 2006390 1000 0 0 -----> We can see high amount of traffic hitting the Sw forward
15 CPU_Q_TOPOLOGY_CONTROL 0 0 0 0
16 CPU_Q_PROTO_SNOOPING 0 0 0 0
17 CPU_Q_DHCP_SNOOPING 0 0 0 0
18 CPU_Q_TRANSIT_TRAFFIC 0 0 0 0
19 CPU_Q_RPF_FAILED 0 0 0 0
20 CPU_Q_MCAST_END_STATION_SERVICE 0 0 0 0
21 CPU_Q_LOGGING 34960 0 0 0
22 CPU_Q_PUNT_WEBAUTH 0 0 0 0
23 CPU_Q_HIGH_RATE_APP 0 0 0 0
24 CPU_Q_EXCEPTION 0 0 0 0
25 CPU_Q_SYSTEM_CRITICAL 0 0 0 0
26 CPU_Q_NFL_SAMPLED_DATA 0 0 0 0
27 CPU_Q_LOW_LATENCY 0 0 0 0
28 CPU_Q_EGR_EXCEPTION 0 0 0 0
29 CPU_Q_FSS 0 0 0 0
30 CPU_Q_MCAST_DATA 0 0 0 0
31 CPU_Q_GOLD_PKT 0 0 0 0
-----------------------------------------------------------------------------
```

在非常短的间隔内多次收集输出：

```
C9500-2#show platform software fed switch active punt rates interfaces 0x42 | in SW_FORWARDING
14 CPU_Q_SW_FORWARDING 2126315 1000 0 0
C9500-2#
C9500-2#show platform software fed switch active punt rates interfaces 0x42 | in SW_FORWARDING
14 CPU_Q_SW_FORWARDING 2128390 1000 0 0
C9500-2#
C9500-2#show platform software fed switch active punt rates interfaces 0x42 | in SW_FORWARDING
14 CPU_Q_SW_FORWARDING 2132295 1000 0 0
C9500-2#
```

这清楚地表明Sw转发队列被阻塞。
一旦您从Te1/0/37删除**ip tcp adjust-mss**命令，或者如果您停止此TCP数据流，从C9500-1到C9200的SSH访问会立即重新建立。

让我们来看看在关闭C9500-2 Te1/0/37之后的SSH会话：

```
C9500-1#ssh -l admin 10.10.20.1
Password:
```

您可以看到SSH访问再次恢复。
因此，您可以在此处将TCP缓慢（SSH访问被阻止）与网络中较高的TCP流量相关联，并进行TCP MSS调整。

要点

1. 只要您的网络中有TCP缓慢（例如文件传输缓慢、TCP相关应用程序的可访问性等），并且您在Catalyst交换机上配置了TCP MSS调整，请确保检查COPP监察器丢弃，以检查网络中是否存在大量TCP流量。
2. 如果在Catalyst交换机上配置了TCP MSS调整，请确保网络中的TCP流量不会超订用COPP监察器速率，否则，网络中会出现TCP相关问题（缓慢、丢包）。

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）。