對Catalyst 9K交換機中由於MSS調整而導致的TCP緩慢問題進行故障排除

目錄

簡介

本檔案將說明Catalyst 9K交換器如何執行TCP MSS調整，以及TCP慢速如何與此功能連結。

有關TCP MSS調整的資訊

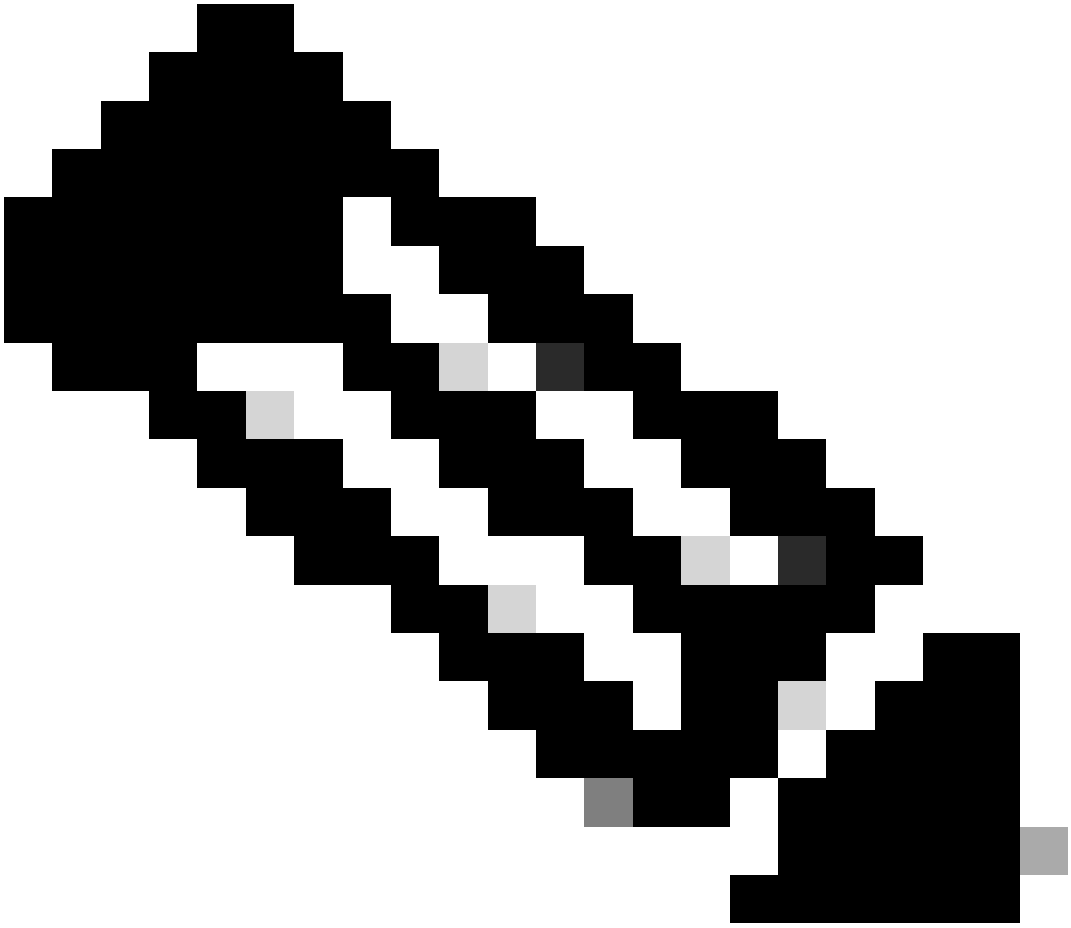
傳輸控制通訊協定(TCP)最大區段大小(MSS)調整功能可設定穿越路由器的暫時性封包的最大區段大小，特別是已設定SYN位元的TCP區段。ip tcp adjust-mss命令在介面配置模式下使用，用於指定SYN資料包的中間路由器上的MSS值，以避免截斷。

主機（通常為PC）與伺服器發起TCP作業階段時，會使用TCP SYN封包中的MSS選項欄位協商IP區段大小。主機上的MTU配置確定MSS欄位的值。PC的NIC的預設MTU值為1500位元組，TCP MSS值為1460（1500位元組- 20位元組IP標頭- 20位元組TCP標頭）。

乙太網路PPP (PPPoE)標準僅支援1492位元組的MTU。

主機和PPPoE MTU大小之間的差異會導致主機和伺服器之間的路由器丟棄1500位元組資料包，並終止透過PPPoE網路的TCP會話。
即使主機上已啟用路徑MTU（可偵測路徑中的正確MTU），系統仍會捨棄作業階段，因為系統管理員有時會停用必須自主機轉送的Internet控制訊息通訊協定(ICMP)錯誤訊息，路徑MTU才能運作。

ip tcp adjust-mss命令透過調整TCP SYN資料包的MSS值來幫助防止TCP會話被丟棄。ip tcp adjust-mss命令僅對透過路由器的TCP連線有效。在大多數情況下，ip tcp adjust-mss命令的max-segment-size引數的最佳值為1452位元組。
此值加上20位元組的IP報頭、20位元組的TCP報頭和8位元組的PPPoE報頭，可增加與乙太網鏈路的MTU大小匹配的1500位元組資料包。



注意：基於TCP MSS調整的資料流在Catalyst 9K交換機中進行軟體交換。本檔案將說明假設TCP MSS調整型流量是軟體交換流量的案例。請參考配置指南，以確認特定硬體/軟體軟體是否切換基於TCP MSS調整的流量。

行為

如前所述，基於TCP MSS調整的流量始終由軟體交換。
這表示如果您嘗試執行TCP調整，則交換器會將TCP流量傳送到CPU以進行MSS修改。

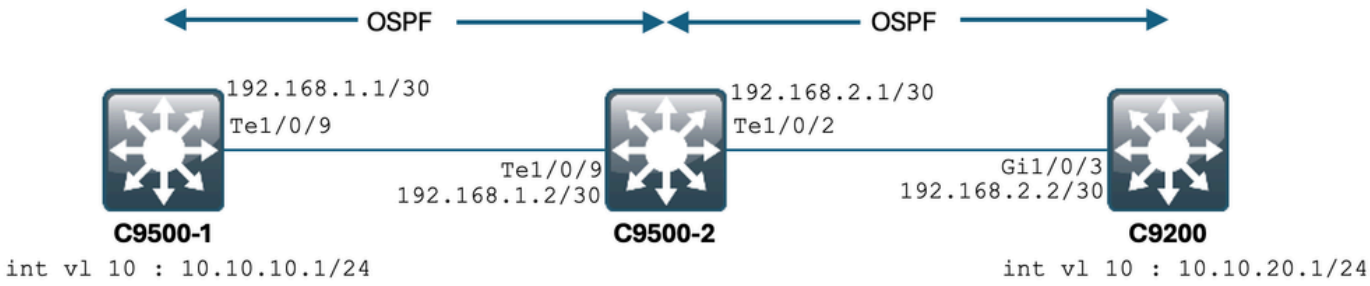
例如，如果修改介面上的TCP MSS值，則在該介面上接收的所有TCP流量都會被傳送到CPU。
然後CPU變更MSS值，並將流量傳送到該TCP封包所經過的所需介面。

因此，如果使用MSS調整有大量的TCP流量，則會超載CPU佇列。
當CPU隊列過載時，控制平面監察器(COPP)會控制流量並丟棄資料包，以保持隊列監察器速率。
這會導致TCP資料包被丟棄。

因此，可能會出現檔案傳輸緩慢、SSH會話建立和Citrix應用程式緩慢（如果使用TCP）等問題。

這裡顯示的是一個實際例子。

拓撲



案例

您將透過SSH從C9500-1進入C9200。
使用C9500-1的VLAN 10 (10.10.10.1)作為源的SSH。
SSH的目的地是C9200的VLAN 20 (10.10.20.1/24)。
SSH是基於TCP的，因此TCP中的任何速度慢也會影響此SSH會話的建立。

在C9500-1和C9200之間有一個中轉L3交換機(C9500-2)。
有兩個中轉/30 L3鏈路，一個在C9500-1和C9500-2之間，另一個在C9500-2和C9200之間。
OSPF用於實現所有三台交換機之間的可接通性，所有/30子網和SVI都通告到OSPF中。
前面顯示的所有IP在它們之間均可訪問。

在C9500-2 Te1/0/9中，完成TCP MSS值的修改。
從C9500-1啟動SSH時，會發生TCP三次握手。
SYN封包命中C9500-2 Te1/0/9（輸入），執行TCP MSS調整。

初始配置和行為

對C9500-2 Te1/0/9（雙向）執行EPC捕獲，並啟動從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#
```

啟動EPC：

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

從C9500-1到C9200啟動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#
```

以下是EPC捕獲的資料包：

```
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 ~
```

您可以看到資料包編號1、2、3中發生了TCP握手。
第1個資料包是SYN資料包。
您可以看到它附帶的MSS值為536。
還發現SYN、ACK資料包（資料包編號2）來自MSS值為536的C9200。
在這裡，MSS值會保持不變，且交換器不會變更。

TCP MSS調整後的行為

以下是C9500-2 Te1/0/9上的TCP MSS調整組態：

```
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.
```

現在，對C9500-2 Te1/0/9（雙向）執行EPC捕獲，然後從C9500-1到C9200啟動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#
```

啟動捕獲，使用SSH從C9500-1到C9200，然後停止捕獲。

以下是CPU捕獲的資料包：

```
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TCP握手發生在資料包編號2、3、4中。
第2個資料包是SYN資料包。
您可以看到它附帶的MSS值為536。
但是，可以看到SYN、ACK資料包（資料包編號3）來自MSS值為512的C9200。
這是因為，當SYN資料包到達C9500-2 Te1/0/9時，它會傳送到C9500-2的CPU，以便將TCP MSS從536修改為512。
C9500-2的CPU將MSS更改為512，並將SYN資料包從Te1/0/2傳送到C9200。
然後所有以下TCP事務使用相同的修改MSS值。

現在，讓我們深入探討SYN資料包如何透過交換機並發生MSS更改。
一旦此SYN資料包到達C9500-2的介面，就會傳送到CPU以修改MSS。
它會先經過FED（您可以在其中捕獲它），然後進入CPU（也可以捕獲它）。

我們首先在C9500-2上捕獲FED Punt。

以下是FED傳送捕獲配置：

```
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
```

正在啟動FED棄權捕獲：

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

從C9500-1到C9200啟動SSH：

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

停止美聯儲棄權捕獲：

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

下面是FED傳輸捕獲的資料包：

```
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
```

您可以看到第2個資料包是從10.10.10.1到10.10.20.1的TCP SYN資料包，來自Te1/0/9。
這裡必須注意'q-no'。您可以看到，它選擇隊列14從FED進入CPU。

您可以在此處看到流量從FED移動到CPU的所有32個隊列：

```
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
```

如您所見，第14號隊列是「Sw forwarding」隊列。
在這種情況下，此隊列由TCP流量用於傳送到CPU。

現在，讓我們在C9500-2上執行CPU（控制平面）捕獲。

以下是CPU捕獲配置：

```
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#
```

開始捕獲後，使用SSH從C9500-1到C9200，然後停止捕獲。

以下是CPU捕獲的資料包：

```
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 ~
```

封包12是進入CPU（傳送）的TCP SYN封包，預設MSS值為536。
封包13是CPU將MSS值修改為512後（插入）發出的TCP SYN封包。

您也可以執行快速CPU偵錯，以便看到發生此變更。

以下是CPU調試配置：

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

從C9500-1到C9200啟動SSH：

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

正在停止CPU調試：

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

檢視調試日誌：

```
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#
```

您可以看到將原始MSS值536調整為512的開銷。

最後，您可以在C9200 Gi1/0/3上執行EPC捕獲，以確認TCP SYN資料包確實帶有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#
```

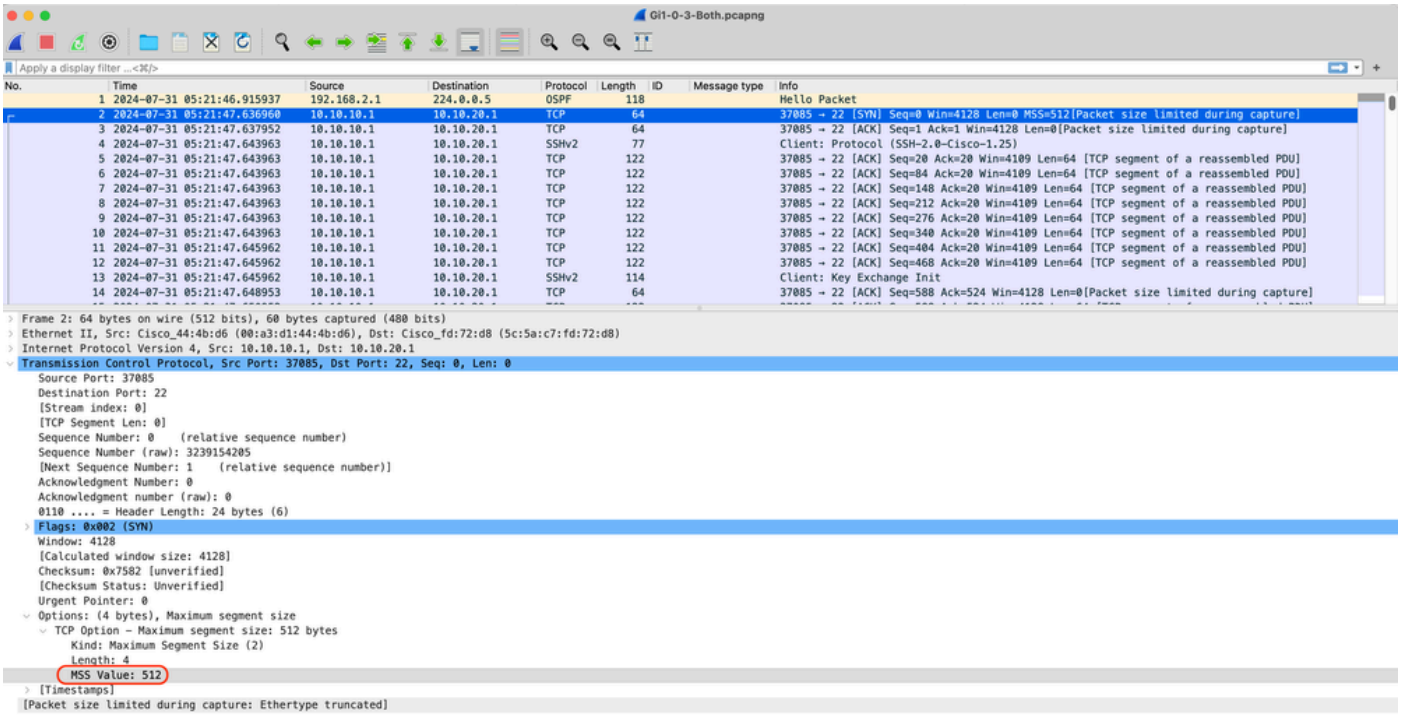
開始捕獲後，使用SSH從C9500-1到C9200，然後停止捕獲。

以下是CPU捕獲的資料包：

```
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，您無法在Wireshark中看到資料包詳細資訊，只有簡要和十六進位制詳細資訊可用。
因此，您可以將早期的資料包導出到快閃記憶體中的pcap檔案。
C9200#mon cap mycap export flash：Gi1-0-3-Both.pcapng
已成功匯出

然後，可以透過TFTP將此檔案複製到本地PC，並在Wireshark中打開該檔案。
這是Wireshark捕獲。



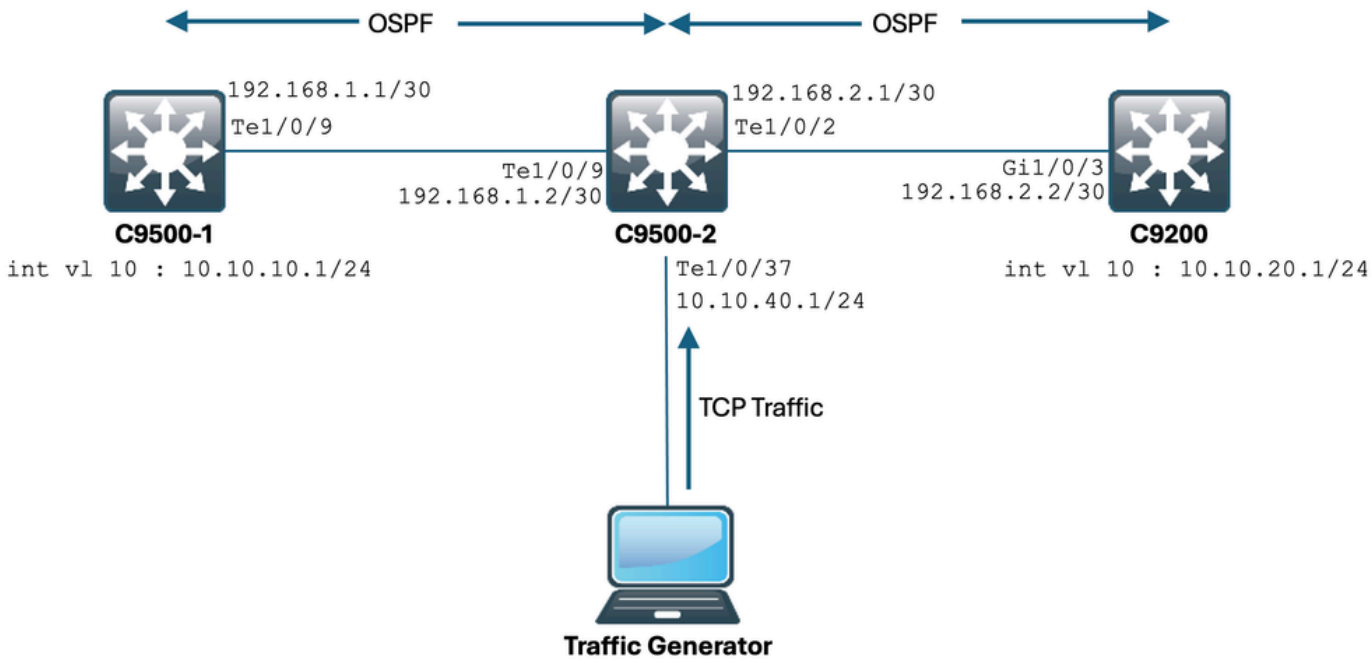
您可以看到SYN封包的TCP MSS值為512。

TCP MSS調整造成大量TCP流量傳輸時速度緩慢

現在，讓我們假設一個網路有多台使用TCP流量的裝置。
例如，它們可以傳輸檔案，或訪問基於TCP的應用（例如Citrix伺服器）。

您將IXIA（流量產生器）連線到C9500-2 Te1/0/37，以高速傳送TCP SYN封包來模擬它。
此IXIA裝置用作網段，其中多個使用者使用基於TCP的應用程式。
您已在Te1/0/37上配置了ip tcp adjust-mss CLI。
這會導致Te1/0/37上接收的所有TCP流量被傳送到C9500-2的CPU。
這反過來又會阻塞C9500-2的COPP監察器的「Sw forwarding」隊列，如本文檔前面所述。
因此，從C9500-1到C9200的SSH會話建立受到影響。
SSH會話未形成、超時，或在延遲後建立。

以下是拓撲的顯示方式：



讓我們來看看它的實際應用。

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
```

現在，您開始從IXIA向Te1/0/37介面傳送大量流量。
讓我們來看看傳入的流量速率：

```
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
```

現在讓我們嘗試從C9500-1到C9200的SSH：

```
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無法通過SSH連線到C9200。
這是因為C9500-1傳送的TCP SYN資料包被「Sw forwarding」隊列丟棄，而該隊列正被來自Te1/0/37的流量所攻擊。

讓我們看一看隊列：

```
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
```

讓我們多次收集輸出，以確保丟棄的計數在問題期間增加：

```
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#
```

如您所見，丟棄的計數增加，並且SSH流量（TCP SYN資料包）在此被丟棄。

現在，如果您不知道流量是透過哪個介面/SVI流入的，您將得到一個特定的命令來幫助。

```
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 feed switch active punt rates interfaces命令提供了負責接收被傳送到CPU的大量流量的介面的清單。
您可以清楚看到此處的Te1/0/37，它是獲取TCP流量的介面。

現在，如果您想檢視到達所有COPP監察器隊列（在較早介面上接收）的流量量，可以使用：
show platform software fed switch active punt rate interfaces<來自以上輸出的IF_ID>

我們來看看：

```
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
------------------------------------------------------------------------------
```

在非常短的間隔內多次收集輸出：

```
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#
```

這清楚地表明Sw轉發隊列被阻塞。
一旦您從Te1/0/37刪除**ip tcp adjust-mss**命令，或者如果您停止此TCP資料流，從C9500-1到C9200的SSH訪問將立即重新建立。

讓我們看一下C9500-2 Te1/0/37關閉後的SSH會話：

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

您可以看到SSH訪問再次恢復。
因此，您可以將因網路中TCP流量過多而導致的TCP緩慢（SSH存取遭封鎖）與TCP MSS調整相關聯。

要點

1. 只要您的網路中有TCP緩慢（例如檔案傳輸緩慢、TCP相關應用程式的存取能力等），而且您在Catalyst交換器上設定了TCP MSS調整，請務必檢查COPP管制器捨棄專案，以檢查網路中是否有大量的TCP流量。
2. 如果您已在Catalyst交換器上設定TCP MSS調整，請確認網路中的TCP流量沒有超額訂閱COPP管制器速率，否則網路中會看到TCP相關的問題（緩慢、封包捨棄）。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。