Catalyst 9K 스위치의 MSS 조정으로 인한 TCP 지연 문제 해결

목차

소개

이 문서에서는 Catalyst 9K Switch가 TCP MSS 조정을 수행하는 방법 및 TCP 속도가 이 기능에 연결되는 방법에 대해 설명합니다.

TCP MSS 조정 정보

TCP(Transmission Control Protocol) MSS(Maximum Segment Size) Adjustment 기능을 사용하면 라우터를 통과하는 임시 패킷, 특히 SYN 비트 세트가 있는 TCP 세그먼트에 대한 최대 세그먼트 크기를 구성할 수 있습니다. 이ip tcp adjust-mss명령은 인터페이스 컨피그레이션 모드에서 SYN 패킷의 중간 라우터에 MSS 값을 지정하여 잘림을 방지하는 데 사용됩니다.

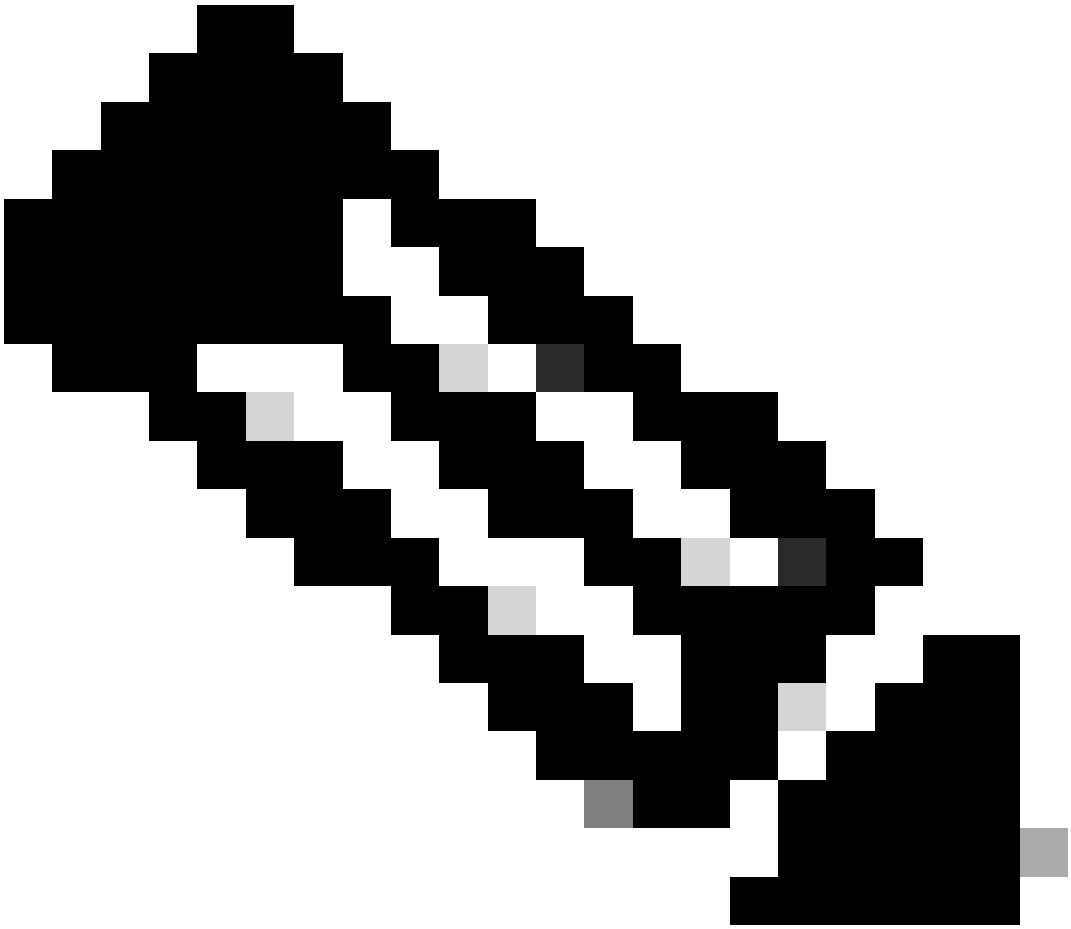
호스트(일반적으로 PC)가 서버와의 TCP 세션을 시작하면 TCP SYN 패킷의 MSS 옵션 필드를 사용하여 IP 세그먼트 크기를 협상합니다. 호스트의 MTU 컨피그레이션에 따라 MSS 필드의 값이 결정됩니다. PC의 NIC에 대한 기본 MTU 값은 1500바이트이며 TCP MSS 값은 1460입니다(1500바이트 - 20바이트 IP 헤더 - 20바이트 TCP 헤더).

PPPoE(PPP over Ethernet) 표준은 1492바이트의 MTU만 지원합니다.

호스트와 PPPoE MTU 크기 간의 차이로 인해 호스트와 서버 간의 라우터가 1500바이트 패킷을 삭제하고 PPPoE 네트워크를 통해 TCP 세션을 종료할 수 있습니다.
경로 MTU(경로 전체에서 올바른 MTU를 탐지함)가 호스트에서 활성화된 경우에도, 시스템 관리자가 경로 MTU가 작동하기 위해 호스트에서 릴레이해야 하는 ICMP(Internet Control Message Protocol) 오류 메시지를 비활성화하는 경우가 있으므로 세션이 삭제될 수 있습니다.

ip tcp adjust-mss 명령은 TCP SYN 패킷의 MSS 값을 조정하여 TCP 세션이 삭제되는 것을 방지합니다. ip tcp adjust-mss 명령은 라우터를 통과하는 TCP 연결에만 적용됩니다. 대부분의 경우 ip tcp

adjust-mss 명령의 max-segment-size 인수에 대한 최적 값은 1452바이트입니다.
이 값에 20바이트 IP 헤더, 20바이트 TCP 헤더, 8바이트 PPPoE 헤더를 더한 값을 더하면 이더넷 링크의 MTU 크기와 일치하는 1500바이트 패킷이 추가됩니다.



참고: TCP MSS 조정 기반 트래픽은 Catalyst 9K Switch에서 소프트웨어 스위칭됩니다. 이 문서에서는 TCP MSS 조정 기반 트래픽이 소프트웨어 스위칭이라고 가정하는 시나리오에 대해 설명합니다. 특정 HW/SW 소프트웨어가 TCP MSS 조정 기반 트래픽을 전환하는지 확인하려면 컨피그레이션 가이드를 참조하십시오.

동작

앞에서 언급한 것처럼 TCP MSS 조정 기반 트래픽은 항상 소프트웨어 스위칭됩니다.
즉, TCP 조정을 시도하고 수행하면 스위치가 MSS 수정을 위해 TCP 트래픽을 CPU로 전송합니다.

예를 들어, 인터페이스에서 TCP MSS 값을 수정할 경우, 해당 인터페이스에서 수신되는 모든 TCP 트래픽이 CPU로 펀트됩니다.

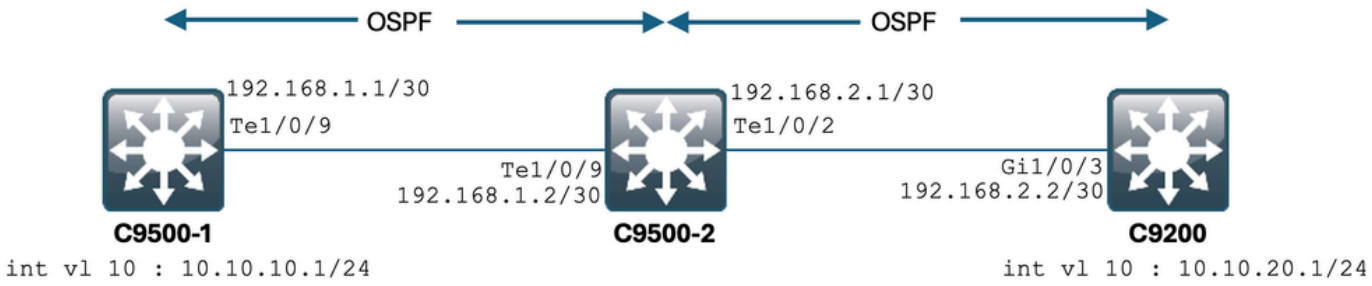
그런 다음 CPU가 MSS 값을 변경하고 트래픽을 해당 TCP 패킷이 향하는 필수 인터페이스로 전송합니다.

이러한 이유로 인해 MSS가 조정되는 TCP 트래픽이 방대할 경우 CPU 대기열이 오버로드됩니다. CPU 큐가 오버로드되면 COPP(Control Plane Policer)는 큐 폴리서 속도를 유지하기 위해 트래픽을 보내고 패킷을 삭제하도록 정책합니다. 이렇게 하면 TCP 패킷이 삭제됩니다.

따라서 파일 전송 속도 저하, SSH 세션 생성, Citrix 애플리케이션 속도 저하(TCP를 사용하는 경우) 등의 문제가 나타납니다.

이것이 어떻게 일어나는지에 대한 실제 예가 여기에 나와 있다.

토폴로지



시나리오

C9500-1에서 C9200으로 SSH를 실행합니다.
C9500-1의 VLAN 10(10.10.10.1)을 소스로 사용하는 SSH.
SSH의 대상은 C9200의 VLAN 20(10.10.20.1/24)입니다.
SSH는 TCP 기반이므로 TCP의 느림도 이 SSH 세션 생성에 영향을 줍니다.

C9500-1과 C9200 사이에 트랜짓 L3 스위치(C9500-2)가 있습니다.
C9500-1과 C9500-2 사이에 하나씩, C9500-2와 C9200 사이에 하나씩 두 개의 트랜짓/30 L3 링크가 있습니다.
OSPF는 3개의 스위치 모두에서 연결성에 사용되며, 모든/30 서브넷 및 SVI가 OSPF에 광고됩니다.
앞에서 설명한 모든 IP는 서로 간에 연결할 수 있습니다.

C9500-2 Te1/0/9에서는 TCP MSS 값 수정이 수행됩니다.
C9500-1에서 SSH가 시작되면 TCP 3-Way 핸드셰이크가 발생합니다.
SYN 패킷이 C9500-2 Te1/0/9(Ingress)에 도달하며, 여기서 TCP MSS 조정이 수행됩니다.

초기 컨피그레이션 및 동작

C9500-2 Te1/0/9(양방향)에서 EPC 캡처를 수행하고 C9500-1에서 C9200으로의 SSH를 시작했습니다.

다음은 EPC 컨피그레이션입니다.

```
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#
```

EPC 시작:

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

C9500-1에서 C9200으로 SSH를 시작합니다.

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

EPC 중단:

```
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#
```

다음은 EPC 캡처 패킷입니다.

```
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 ~
```

TCP 핸드셰이크가 패킷 번호 1,2,3에서 발생하는 것을 확인할 수 있습니다.
Packet No.1은 SYN 패킷입니다.
MSS 값이 536으로 나와 있습니다.
SYN, ACK 패킷(패킷 번호 2)도 MSS 값이 536인 C9200에서 오는 것으로 보입니다.
여기서 MSS 값은 그대로 유지되며 스위치에 의해 변경되지 않습니다.

TCP MSS 조정 후 동작

다음은 C9500-2 Te1/0/9의 TCP MSS 조정 컨피그레이션입니다.

```
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.
```

이제 C9500-2 Te1/0/9(양방향)에서 EPC 캡처를 수행하고 C9500-1에서 C9200으로 SSH를 시작합니다.

다음은 EPC 컨피그레이션입니다.

```
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#
```

캡처를 시작하고 C9500-1에서 C9200으로 SSH를 수행한 후 캡처를 중지합니다.

다음은 CPU 캡처 패킷입니다.

```
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 ~
```

2,3,4 패킷에서 발생하는 TCP 핸드셰이크를 볼 수 있습니다.
패킷 No.2는 SYN 패킷입니다.
MSS 값이 536으로 나와 있습니다.
그러나 SYN, ACK 패킷(패킷 번호 3)은 MSS 값이 512인 C9200에서 오는 것으로 보입니다.
이는 SYN 패킷이 C9500-2 Te1/0/9에 도달하면 536에서 512로 TCP MSS 수정을 위해 C9500-2의 CPU로 전송되기 때문입니다.
C9500-2의 CPU는 MSS를 512로 변경하고 SYN 패킷을 Te1/0/2에서 C9200으로 전송합니다.
그런 다음 모든 후속 TCP 트랜잭션은 동일한 수정된 MSS 값을 사용합니다.

이제 SYN 패킷이 스위치를 통과하여 MSS 변경이 발생하는 방식을 자세히 살펴보겠습니다.
이 SYN 패킷이 C9500-2의 인터페이스에 도달하면 MSS 수정을 위해 CPU로 전송됩니다.
먼저 FED(캡처할 수 있는 위치)를 거친 다음 CPU(캡처할 수 있는 위치)로 이동합니다.

먼저 C9500-2의 FED Punt 포착을 들어보겠습니다.

다음은 FED punt 캡처 컨피그레이션입니다.

```
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
```

FED punt capture 시작:

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

C9500-1에서 C9200으로 SSH를 시작합니다.

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

FED punt 캡처 중지:

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

다음은 FED punt 캡처 패킷입니다.

```
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
```

패킷 2는 Te1/0/9에서 들어오는 10.10.10.1~10.10.20.1의 TCP SYN 패킷입니다.
여기서 'q-no'는 중요합니다. FED에서 CPU로 이동하기 위해 Queue No. 14를 선택하는 것을 볼 수 있습니다.

FED에서 CPU로 이동하는 트래픽을 위해 존재하는 32개의 모든 대기열을 볼 수 있습니다.

```
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
```

오버헤드를 볼 수 있듯이 14번 대기열이 'Sw 전달' 대기열입니다.
이 경우 이 대기열은 TCP 트래픽에서 CPU에 할당되기 위해 사용됩니다.

이제 C9500-2의 CPU(Control-Plane) 캡처를 살펴보겠습니다.

다음은 CPU 캡처 컨피그레이션입니다.

```
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#
```

C9500-1에서 C9200으로 SSH 캡처를 시작하고 캡처를 중지합니다.

다음은 CPU 캡처 패킷입니다.

```
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 ~
```

패킷 번호 12는 CPU(punt)로 들어오는 TCP SYN 패킷이며, 기본 MSS 값은 536입니다.
패킷 번호 13은 MSS 값을 512로 수정한 후 CPU(삽입)에서 전송한 TCP SYN 패킷입니다.

또한 빠른 CPU 디버그를 통해 이러한 변경 사항을 확인할 수 있습니다.

다음은 CPU 디버그 컨피그레이션입니다.

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

C9500-1에서 C9200으로 SSH를 시작합니다.

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

CPU 디버그 중지:

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

디버그에 대한 로그를 확인합니다.

```
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#
```

Original MSS 값 536이 512로 조정되는 오버헤드를 확인할 수 있습니다.

마지막으로, TCP SYN 패킷이 실제로 MSS 512와 함께 제공됨을 확인하기 위해 C9200 Gi1/0/3에서 EPC 캡처를 수행할 수 있습니다.

다음은 EPC 컨피그레이션입니다.

```
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#
```

C9500-1에서 C9200으로 SSH 캡처를 시작하고 캡처를 중지합니다.

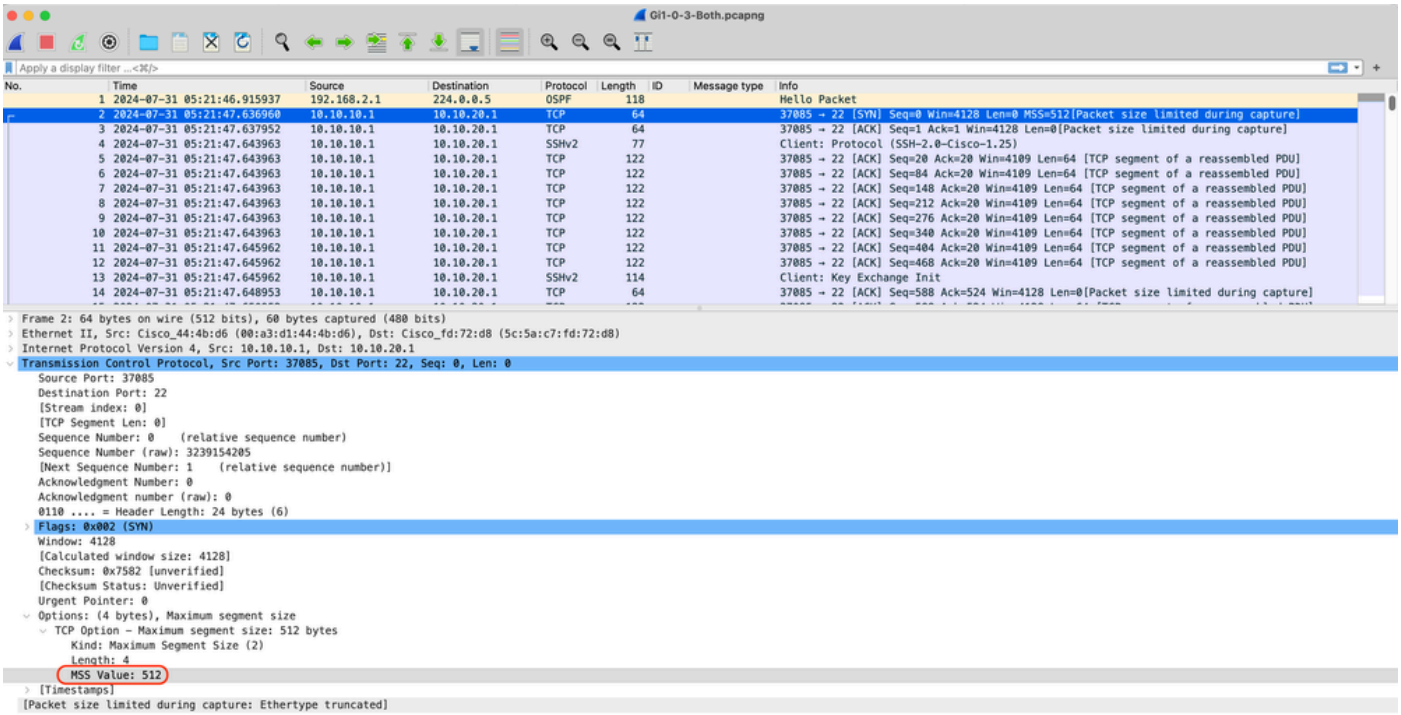
다음은 CPU 캡처 패킷입니다.

```
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 ~
```

C9200에서는 Wireshark와 같은 패킷 세부사항을 볼 수 없으며 간략한 및 16진수 세부사항만 사용할 수 있습니다.
따라서 플래시의 pcap 파일로 이전 패킷을 내보낼 수 있습니다.
C9200#mon cap mycap export flash:Gi1-0-3-Both.pcapng

성공적으로 내보냄

그런 다음 TFTP를 통해 이 파일을 로컬 PC에 복사하고 Wireshark에서 파일을 열 수 있습니다.
여기 Wireshark가 잡혔습니다.



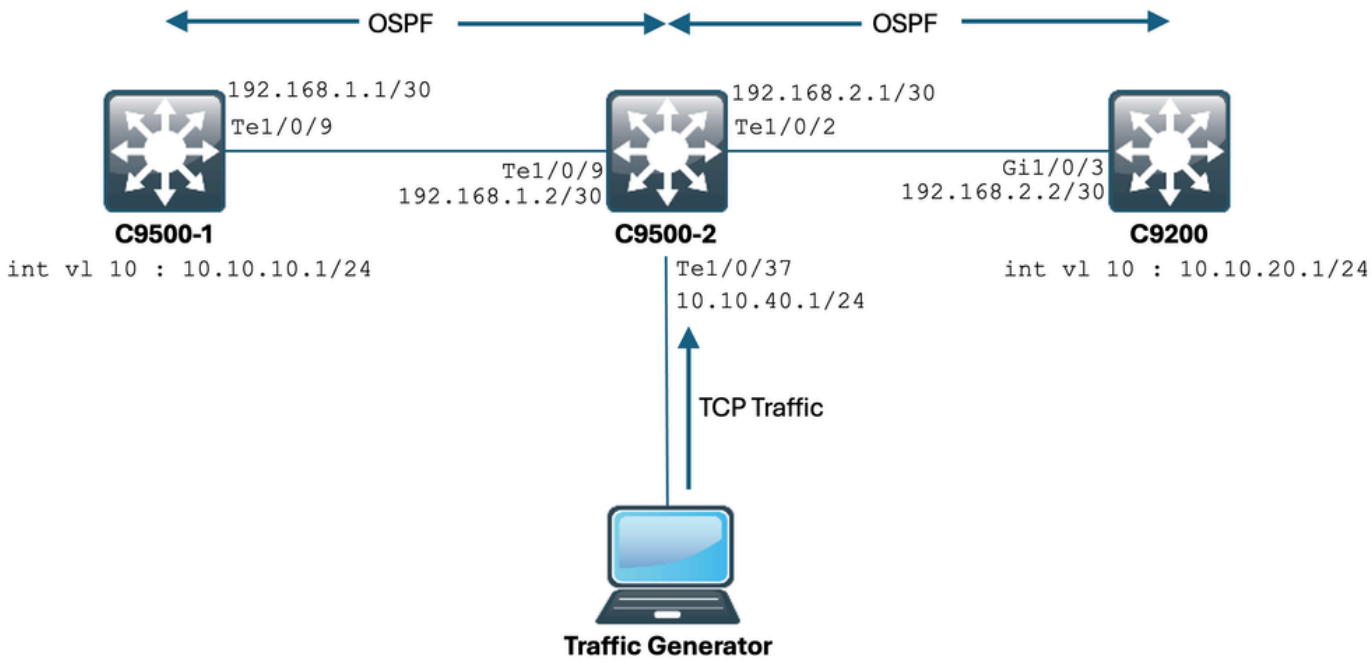
SYN 패킷의 TCP MSS 값이 512임을 확인할 수 있습니다.

TCP 트래픽 양이 많은 동안 느려지는 TCP MSS 조정

이제 네트워크에 TCP 트래픽을 사용하는 여러 디바이스가 있다고 가정하겠습니다.
예를 들어, 파일을 전송하거나 TCP 기반 애플리케이션(예: Citrix Server)에 액세스할 수 있습니다.

IXIA(트래픽 생성기)를 C9500-2 Te1/0/37에 연결하여 높은 속도로 TCP SYN 패킷을 전송하여 시뮬레이션했습니다.
이 IXIA 디바이스는 여러 사용자가 TCP 기반 애플리케이션을 사용하는 네트워크 세그먼트 역할을 합니다.
Te1/0/37에서 ip tcp adjust-mss CLI를 구성했습니다.
이렇게 하면 Te1/0/37에서 수신되는 모든 TCP 트래픽이 C9500-2의 CPU에 기록됩니다.
그러면 앞서 문서에서 언급한 대로 C9500-2의 COPP 폴리서의 'Sw 전달' 대기열이 닫힙니다.
따라서 C9500-1에서 C9200으로의 SSH 세션 설정이 영향을 받습니다.
SSH 세션이 형성되지 않고 시간이 초과되거나 지연 후 설정됩니다.

토폴로지의 모양은 다음과 같습니다.



이것을 실제로 봅시다.

C9500-2 Te1/0/37의 구성은 다음과 같습니다.

```
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
```

이제 IXIA에서 Te1/0/37 인터페이스로 방대한 트래픽을 전송하기 시작합니다.
수신 트래픽 속도를 살펴보겠습니다.

```
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
```

이제 C9500-1에서 C9200으로 SSH를 시도해보겠습니다.

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

C9500-1에서 C9200에 SSH를 적용할 수 없었음을 분명히 알 수 있습니다.
이는 C9500-1에서 전송 중인 TCP SYN 패킷이 'Sw 전달' 대기열에 의해 삭제되었기 때문이며,
Te1/0/37의 트래픽이 폭주하고 있습니다.

대기열을 살펴보겠습니다.

```
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
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
```

문제가 발생하는 동안 삭제 횟수가 증가하도록 출력을 여러 번 수집하겠습니다.

```
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#
```

보시는 바와 같이 삭제된 수는 증가하고 있으며, SSH 트래픽(TCP SYN 패킷)은 여기에서 삭제됩니다.

이제 어떤 인터페이스/SVI를 통해 이러한 트래픽 유입이 발생하는지 모르는 경우 특정 명령을 사용하여 문제를 해결할 수 있습니다.

```
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 rate interfaces 명령은 CPU로 펀트되는 막대한 양의 트래픽을 수신하는 데 책임이 있는 인터페이스의 목록을 제공합니다.
TCP 트래픽을 가져오는 데 사용되는 인터페이스인 Te1/0/37을 여기서 확인할 수 있습니다.

이제 (이전 인터페이스에서 수신 중인) 모든 COPP 폴리서 대기열에 도달하는 트래픽 양을 보려면 다음을 사용할 수 있습니다.
show platform software fed switch active punt rate interfaces <IF_ID from the above output>

살펴보겠습니다.

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

매우 짧은 간격으로 출력을 여러 번 수집합니다.

```
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#
```

이는 소프트웨어 전달 대기열이 차단되었음을 명확히 보여줍니다.

Te1/0/37에서 명령을 **ip tcp adjust-mss** 제거하거나 이 TCP 트래픽을 중지하면 C9500-1에서 C9200으로의 SSH 액세스가 즉시 다시 설정됩니다.

C9500-2 Te1/0/37을 종료한 후 SSH 세션을 살펴보겠습니다.

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

SSH 액세스가 다시 복원되는 것을 확인할 수 있습니다.
따라서 네트워크에서 TCP 트래픽의 양이 많기 때문에 여기에서 TCP 느림(SSH 액세스가 차단됨)을 TCP MSS 조정과 상호 연결할 수 있습니다.

중요 사항

1. 네트워크에서 파일 전송 속도, TCP 관련 애플리케이션에 대한 액세스 가능성 등 TCP 속도가 느려지고 Catalyst Switch에서 TCP MSS 조정이 구성된 경우, 네트워크에 많은 양의 TCP 트래픽이 있는지 확인하기 위해 COPP Policer 삭제를 확인해야 합니다.
2. Catalyst Switch에서 TCP MSS 조정을 구성한 경우 네트워크의 TCP 트래픽이 COPP 폴리서 속도를 오버서브스크립션하지 않는지 확인합니다. 그렇지 않으면 네트워크에서 TCP 관련 문제(느림, 패킷 삭제)가 나타납니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.