Fehlerbehebung bei TCP-Langsamkeitsproblemen aufgrund von MSS-Anpassung in Catalyst Switches der Serie 9000

Inhalt

Einleitung

In diesem Dokument wird beschrieben, wie ein Catalyst 9000-Switch die TCP-MSS-Anpassung vornimmt und wie die TCP-Langsamkeit mit dieser Funktion verknüpft ist.

Informationen zur TCP-MSS-Anpassung

Die Funktion zur Anpassung der maximalen Segmentgröße (Maximum Segment Size, MSS) des Transmission Control Protocol (TCP) ermöglicht die Konfiguration der maximalen Segmentgröße für transiente Pakete, die einen Router durchlaufen, insbesondere TCP-Segmente mit dem SYN-Bitsatz. Der`ip tcp adjust-mss`Befehl wird im Schnittstellenkonfigurationsmodus verwendet, um den MSS-Wert auf dem Zwischenrouter der SYN-Pakete anzugeben, um eine Kürzung zu vermeiden.

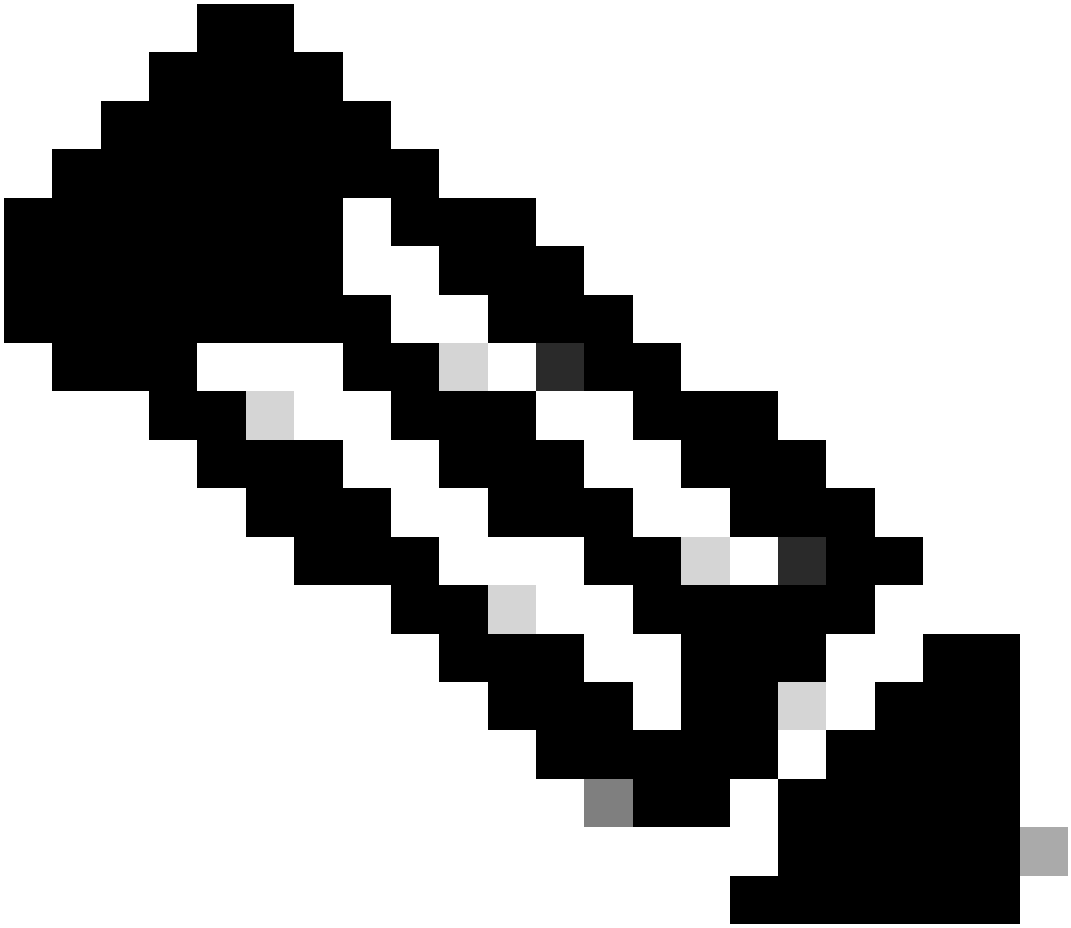
Wenn ein Host (in der Regel ein PC) eine TCP-Sitzung mit einem Server einleitet, handelt er die IP-Segmentgröße mithilfe des MSS-Optionsfelds im TCP-SYN-Paket aus. Die MTU-Konfiguration auf dem Host bestimmt den Wert des MSS-Felds. Der MTU-Standardwert für eine Netzwerkkarte des PCs beträgt 1.500 Byte mit einem TCP-MSS-Wert von 1.460 (1.500 Byte - 20 Byte IP-Header - 20 Byte TCP-Header).

Der PPP over Ethernet (PPPoE)-Standard unterstützt eine MTU von nur 1.492 Byte.

Die Unterschiede zwischen Host und PPPoE-MTU-Größe können dazu führen, dass der Router zwischen Host und Server 1500-Byte-Pakete verwirft und TCP-Sitzungen über das PPPoE-Netzwerk beendet.
Selbst wenn die Pfad-MTU (die die richtige MTU auf dem Pfad erkennt) auf dem Host aktiviert ist,

können Sitzungen verworfen werden, da Systemadministratoren manchmal die ICMP-Fehlermeldungen (Internet Control Message Protocol) deaktivieren, die vom Host weitergeleitet werden müssen, damit die Pfad-MTU funktioniert.

Der Befehl ip tcp adjust-mss verhindert das Verwerfen von TCP-Sitzungen, indem der MSS-Wert der TCP-SYN-Pakete angepasst wird. Der Befehl ip tcp adjust-mss ist nur für TCP-Verbindungen wirksam, die den Router passieren. In den meisten Fällen beträgt der optimale Wert für das Argument max-segment-size des Befehls ip tcp adjust-mss 1452 Byte.
Dieser Wert plus der 20-Byte-IP-Header, der 20-Byte-TCP-Header und der 8-Byte-PPPoE-Header addieren sich zu einem 1500-Byte-Paket, das der MTU-Größe für die Ethernet-Verbindung entspricht.



Hinweis: Datenverkehr, der auf TCP-MSS-Anpassung basiert, wird in Catalyst Switches der Serie 9000 über Software geswitcht. In diesem Dokument werden Szenarien erläutert, bei denen davon ausgegangen wird, dass der auf der TCP-MSS-Anpassung basierende Datenverkehr softwarebasiert ist. Überprüfen Sie im Konfigurationshandbuch, ob eine bestimmte HW/SW-Software den TCP-MSS-Anpassungsdatenverkehr umschaltet.

Verhalten

Wie bereits erwähnt, wird der auf der TCP-MSS-Anpassung basierende Datenverkehr immer über Software weitergeleitet.
Wenn Sie also versuchen, eine TCP-Anpassung durchzuführen, sendet der Switch den TCP-Datenverkehr zur MSS-Änderung an die CPU.

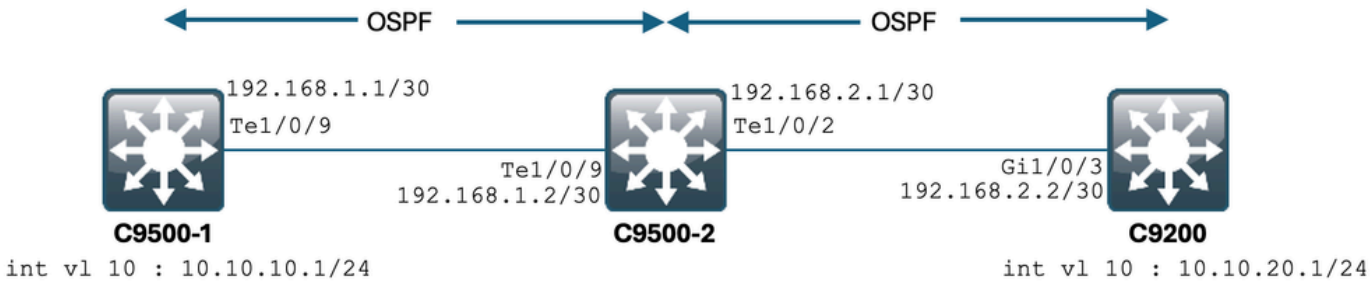
Wenn Sie beispielsweise den TCP-MSS-Wert einer Schnittstelle ändern, wird der gesamte TCP-Datenverkehr, der über diese Schnittstelle empfangen wird, an die CPU gesendet.
Die CPU ändert dann den MSS-Wert und sendet den Datenverkehr an die erforderliche Schnittstelle, auf die das TCP-Paket geleitet wurde.

Aus diesem Grund wird die CPU-Warteschlange überlastet, wenn bei der MSS-Anpassung ein enormer TCP-Datenverkehr auftritt.
Wenn eine CPU-Warteschlange überlastet ist, führt der Control Plane Policer (COPP) Richtlinien für den Datenverkehr durch und verwirft Pakete, um die Warteschlangenüberwachungsrate beizubehalten. Dadurch werden die TCP-Pakete verworfen.

Daher treten Probleme wie Langsamkeit bei der Dateiübertragung, Erstellung von SSH-Sitzungen und Langsamkeit von Citrix-Anwendungen (bei Verwendung von TCP) auf.

Ein Beispiel aus dem wirklichen Leben, wie das passiert, sehen Sie hier.

Topologie



Szenario

Sie wechseln vom C9500-1 zu SSH auf dem C9200.
SSH mit VLAN 10 (10.10.10.1) des C9500-1 als Quelle.
Das Ziel des SSH ist VLAN 20 des C9200 (10.10.20.1/24).
SSH ist TCP-basiert, daher wirkt sich eine verlangsamte TCP-Übertragung auch auf die Erstellung von SSH-Sitzungen aus.

Es gibt einen L3-Transit-Switch (C9500-2) zwischen C9500-1 und C9200.
Es gibt zwei L3-Transit-/30-Verbindungen, eine zwischen C9500-1 und C9500-2 und eine

zwischen C9500-2 und C9200.
OSPF wird für die Erreichbarkeit über alle drei Switches hinweg verwendet, und alle/30 Subnetze und SVIs werden in OSPF angekündigt.
Alle zuvor gezeigten IP-Adressen sind untereinander erreichbar.

In C9500-2 Te1/0/9 wird der TCP-MSS-Wert geändert.
Wenn das SSH vom C9500-1 initiiert wird, tritt ein TCP-3-Wege-Handshake auf.
Das SYN-Paket erreicht den C9500-2 Te1/0/9 (Ingress), wo die TCP-MSS-Anpassung durchgeführt wird.

Erstkonfiguration und Verhalten

Ein EPC-Capture auf C9500-2 Te1/0/9 (beide Richtungen) wurde durchgeführt, und SSH von C9500-1 auf C9200 wurde gestartet.

Hier ist die EPC-Konfiguration:

```
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#
```

Starten des EPC:

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

Starten von SSH von C9500-1 bis C9200:

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

```
Password:
```

Stoppen des 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#
```

Hier sind die vom EPC erfassten Pakete:

```
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 ~
```

Sie können den TCP-Handshake in Paket 1,2,3 sehen.
Paket Nr.1 ist das SYN-Paket.
Wie Sie sehen, ist der MSS-Wert 536.
Das SYN, ACK-Paket (Paket Nr. 2) wird ebenfalls vom C9200 mit einem MSS-Wert von 536 empfangen.
In diesem Fall bleibt der MSS-Wert intakt und wird vom Switch nicht geändert.

Verhalten nach TCP-MSS-Anpassung

Dies ist die TCP-MSS-Anpassungskonfiguration für C9500-2 Te1/0/9:

```
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.
```

Nehmen Sie jetzt eine EPC-Erfassung für C9500-2 Te1/0/9 (beide Richtungen), und starten Sie SSH von C9500-1 bis C9200.

Hier ist die EPC-Konfiguration:

```
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#
```

Starten Sie die Erfassung, SSH von C9500-1 bis C9200, und beenden Sie die Erfassung.

Die CPU-Pakete wurden erfasst:

```
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 ~
```

Sie können den TCP-Handshake in den Paketen 2, 3 und 4 sehen.
Paket Nr.2 ist das SYN-Paket.
Wie Sie sehen, ist der MSS-Wert 536.
Das SYN-ACK-Paket (Paket Nr. 3) stammt jedoch vom C9200 mit einem MSS-Wert von 512.
Wenn das SYN-Paket den C9500-2 Te1/0/9 erreicht, wird es an die CPU des C9500-2 gesendet, um die TCP-MSS von 536 auf 512 zu ändern.
Die CPU des C9500-2 ändert die MSS in 512 und sendet das SYN-Paket aus Te1/0/2 an C9200.
Anschließend verwenden alle folgenden TCP-Transaktionen den gleichen geänderten MSS-Wert.

Sehen wir uns nun genauer an, wie das SYN-Paket den Switch passiert, und wie sich die MSS ändert.
Sobald dieses SYN-Paket die Schnittstelle des C9500-2 erreicht, wird es zur MSS-Änderung an die CPU gesendet.
Es geht zuerst durch die FED (wo Sie es erfassen können), und dann zur CPU (wo Sie es auch erfassen können).

Lassen Sie uns zuerst einen FED Punt-Test auf C9500-2 machen.

Hier ist die Konfiguration der FED-Punkterfassung:

```
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
```

Starten der FED-Punkterfassung:

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

Starten von SSH von C9500-1 bis C9200:

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

Stoppen der FED-Puntaufnahme:

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

Und hier sind die FED-Bot-erfassten Pakete:

```
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
```

Sie können sehen, dass Paket Nr. 2 das TCP-SYN-Paket von 10.10.10.1 bis 10.10.20.1 ist, das von Te1/0/9 eingeht.
Die "q-no" ist hier wichtig zu beachten. Wie Sie sehen, wählt er die Warteschlange Nr. 14 aus, um von der FED zur CPU zu wechseln.

Hier sehen Sie alle 32 Warteschlangen, die vorhanden sind, damit der Datenverkehr vom FED zur CPU fließt:

```
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
```

Wie Sie sehen, ist Warteschlange Nr. 14 die 'Sw-Forwarding'-Warteschlange.
In diesem Fall wird diese Warteschlange vom TCP-Datenverkehr verwendet, um an die CPU weitergeleitet zu werden.

Nehmen wir nun eine CPU-Erfassung (Kontrollebene) für C9500-2.

Die CPU-Erfassungskonfiguration lautet wie folgt:

```
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#
```

Sie starten die Erfassung, SSH von C9500-1 bis C9200, und stoppen die Erfassung.

Die CPU-Pakete wurden erfasst:

```
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 ~
```

Paket Nr. 12 ist das TCP SYN Paket, das in die CPU (punt) kommt, mit dem Standard MSS Wert 536.
Paket Nr. 13 ist das TCP SYN-Paket, das von der CPU gesendet wird (inject), nachdem der MSS-Wert auf 512 geändert wurde.

Sie können auch eine kurze CPU-Fehlersuche durchführen, um diese Änderung zu sehen.

Die CPU-Debug-Konfiguration sieht wie folgt aus:

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

Starten von SSH von C9500-1 bis C9200:

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

Beenden der CPU-Fehlersuche:

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

Protokolle für die Fehlersuche:

```
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#
```

Sie können sehen, dass der ursprüngliche MSS-Wert von 536 auf 512 angepasst wird.

Schließlich können Sie eine EPC-Erfassung für C9200 Gi1/0/3 durchführen, um zu bestätigen, dass das TCP-SYN-Paket tatsächlich mit einer MSS von 512 eingeht.

Hier ist die EPC-Konfiguration:

```
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#
```

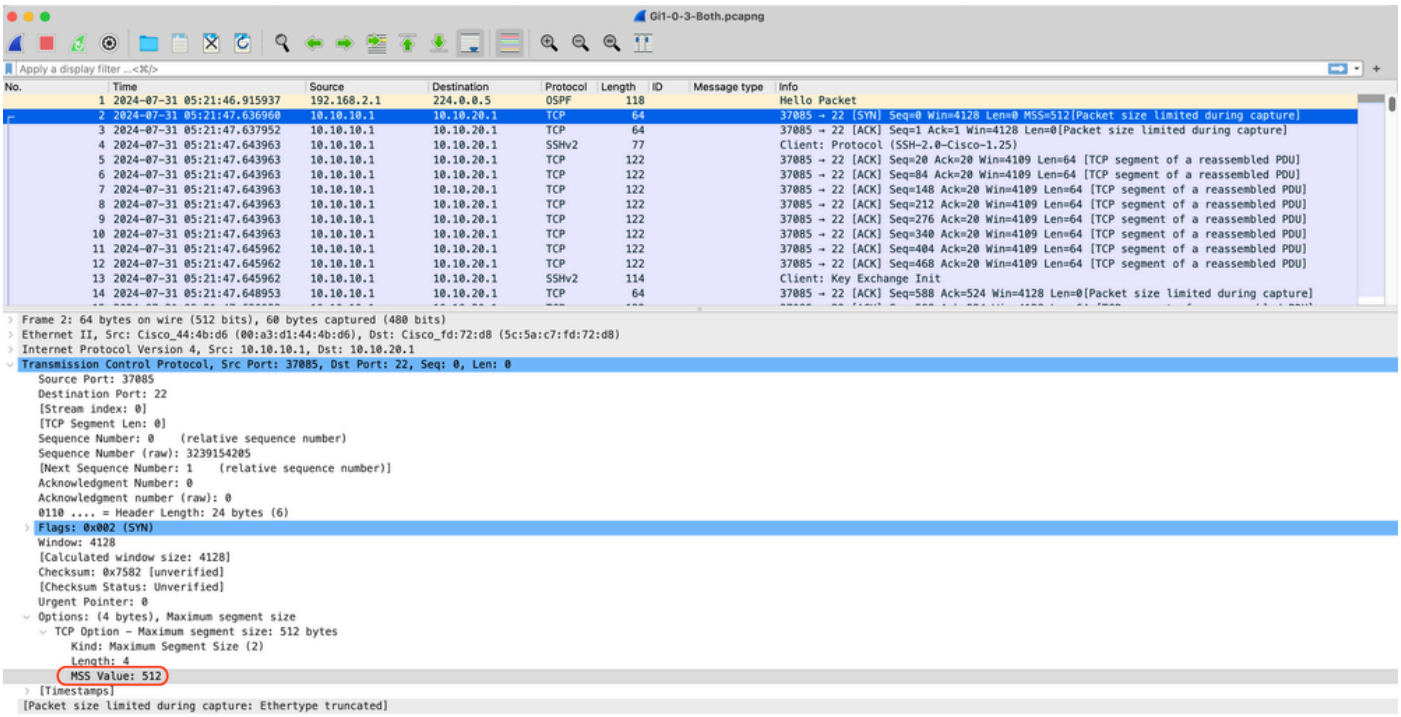
Sie starten die Erfassung, SSH von C9500-1 bis C9200, und stoppen die Erfassung.

Die CPU-Pakete wurden erfasst:

```
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 ~
```

In C9200 können Sie die Paketdetails nicht wie in Wireshark anzeigen. Es sind nur die kurzen und Hexadezimaldetails verfügbar.
Daher können Sie die früheren Pakete in eine pcap-Datei im Flash exportieren.
C9200#mon cap mycap export flash:Gi1-0-3-Both.pcap
Erfolgreich exportiert

Anschließend können Sie diese Datei über TFTP auf Ihren lokalen PC kopieren und sie in Wireshark öffnen.
Hier ist die Erfassung von Wireshark.



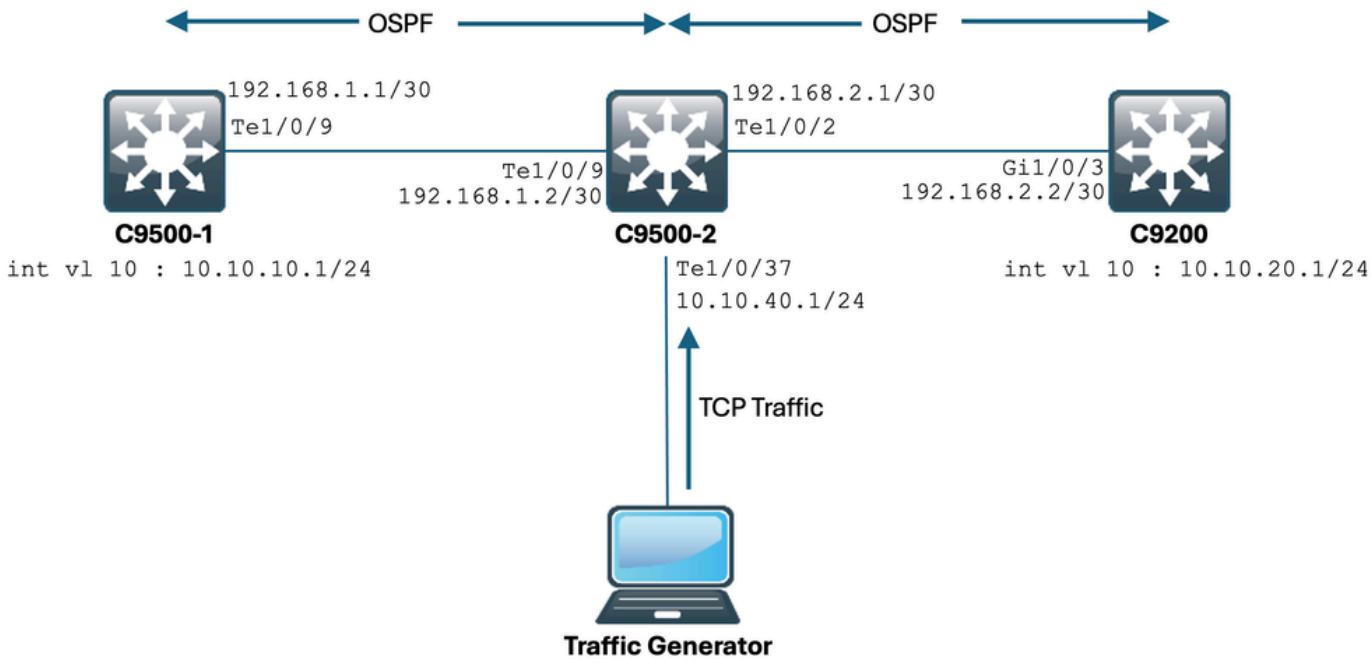
Sie können sehen, dass der TCP-MSS-Wert des SYN-Pakets 512 ist.

TCP-MSS-Anpassung verursacht Verzögerung beim enormen TCP-Datenverkehr

Nehmen wir nun an, ein Netzwerk hat mehrere Geräte, die TCP-Datenverkehr verwenden. Beispielsweise können sie Dateien übertragen oder auf eine TCP-basierte Anwendung (wie einen Citrix Server) zugreifen.

Sie haben es simuliert, indem Sie einen IXIA (Traffic Generator) an C9500-2 Te1/0/37 anschließen und TCP-SYN-Pakete mit hoher Rate senden.
Dieses IXIA-Gerät fungiert als Netzwerksegment, in dem mehrere Benutzer TCP-basierte Anwendungen verwenden.
Sie haben die CLI ip tcp adjust-mss auf Te1/0/37 konfiguriert.
Dadurch wird der gesamte TCP-Datenverkehr, der auf Te1/0/37 empfangen wird, auf die CPU von C9500-2 geleitet.
Dadurch wird wiederum die Warteschlange "Sw forwarding" des COPP Policer beim C9500-2 gekürzt, wie bereits zuvor in diesem Dokument erwähnt.
Dies hat Auswirkungen auf den SSH-Sitzungsaufbau von C9500-1 bis C9200.
Entweder wird die SSH-Sitzung nicht gebildet, und es tritt ein Timeout auf, oder die Sitzung wird verzögert aufgebaut.

Die Topologie sieht wie folgt aus:



Sehen wir uns das in Aktion an.

Die Konfiguration für C9500-2 Te1/0/37 sieht wie folgt aus:

```
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
```

Jetzt beginnen Sie, enormen Datenverkehr von IXIA in die Te1/0/37-Schnittstelle zu senden. Sehen wir uns nun die eingehende Datenverkehrsrate an:

```
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
```

Versuchen wir jetzt SSH von C9500-1 auf C9200:

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

Wie Sie sehen, war der C9500-1 nicht in der Lage, per SSH auf den C9200 zuzugreifen.
Der Grund hierfür ist, dass das vom C9500-1 gesendete TCP-SYN-Paket von der Warteschlange "Sw forwarding" verworfen wurde, die mit dem Datenverkehr vom Te1/0/37 bombardiert wird.

Schauen wir uns die Warteschlange an:

```
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
```

Lassen Sie uns die Ausgabe mehrmals erfassen, um sicherzustellen, dass die Anzahl der verlorenen Nachrichten während des Problems steigt:

```
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#
```

Wie Sie sehen, nimmt die Anzahl der Verworfenen zu, und der SSH-Datenverkehr (TCP SYN-Paket) wird hier verworfen.

Wenn Sie nicht wissen, über welche Schnittstelle/SVI Sie diesen Datenverkehrszufluss erhalten, steht Ihnen ein spezieller Befehl zur Verfügung.

```
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#
```

Der Befehl show platform software fed switch active punt rates interfaces gibt uns die Liste der Schnittstellen, die verantwortlich sind für den Empfang riesiger Datenmengen, die an die CPU gesendet werden.
Hier sehen Sie deutlich Te1/0/37, die Schnittstelle, über die Sie den TCP-Datenverkehr empfangen.

Wenn Sie nun den Umfang des Datenverkehrs sehen möchten, der alle COPP Policer-Warteschlangen erreicht (der über die frühere Schnittstelle empfangen wird), können Sie Folgendes verwenden:
show platform software fed switch active puntrates interfaces <IF_ID from the above output>

Werfen wir einen Blick darauf:

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

Sammeln Sie die Ausgabe mehrmals in sehr kurzen Intervallen:

```
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#
```

Dies zeigt deutlich, dass die SW-Weiterleitungswarteschlange blockiert ist.
Wenn Sie den **ip tcp adjust-mss** Befehl aus Te1/0/37 entfernen oder diesen TCP-Datenverkehr stoppen, wird der SSH-Zugriff von C9500-1 auf C9200 sofort wiederhergestellt.

Sehen wir uns nun die SSH-Sitzung nach dem Herunterfahren von C9500-2 Te1/0/37 an:

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

Sie können sehen, dass der SSH-Zugriff wieder hergestellt wurde.
Daher können Sie die TCP-Langsamkeit hier (SSH-Zugriff blockiert) aufgrund des hohen TCP-Datenverkehrs im Netzwerk mit einer TCP-MSS-Anpassung korrelieren.

Wichtige Punkte

1. Wenn Sie eine langsame TCP-Übertragung in Ihrem Netzwerk haben, z. B. eine langsame Dateiübertragung, Zugriff auf TCP-bezogene Anwendungen usw., und wenn Sie eine TCP-MSS-Anpassung auf einem Catalyst Switch konfiguriert haben, stellen Sie sicher, dass die COPP-Policer-Drops überprüft werden, um zu überprüfen, ob eine große Menge an TCP-Datenverkehr im Netzwerk vorhanden ist.
2. Wenn Sie die TCP-MSS-Anpassung auf einem Catalyst Switch konfiguriert haben, stellen Sie sicher, dass der TCP-Datenverkehr in Ihrem Netzwerk nicht die COPP Policer-Rate überbelegt, da andernfalls TCP-bezogene Probleme (Langsamkeit, Paketverluste) in Ihrem Netzwerk auftreten.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.