Solucionar problemas de lentidão do TCP devido ao ajuste do MSS nos Switches Catalyst 9K

Contents

Introdução

Informações sobre o Ajuste TCP MSS

Comportamento

Topologia

Cenário

Configuração e comportamento iniciais

Comportamento após ajuste de TCP MSS

Ajuste TCP MSS causando lentidão durante uma quantidade enorme de tráfego TCP

Pontos importantes

Introdução

Este documento descreve como um Switch Catalyst 9K faz o ajuste de MSS TCP e como a lentidão do TCP é vinculada a esse recurso.

Informações sobre o Ajuste TCP MSS

O recurso de Ajuste de Tamanho Máximo de Segmento (MSS) do Transmission Control Protocol (TCP) permite a configuração do tamanho máximo de segmento para pacotes transitórios que atravessam um roteador, especificamente segmentos TCP com o bit SYN definido. O comando`ip tcp adjust-mss`é usado no modo de configuração de interface para especificar o valor de MSS no roteador intermediário dos pacotes SYN para evitar truncamento.

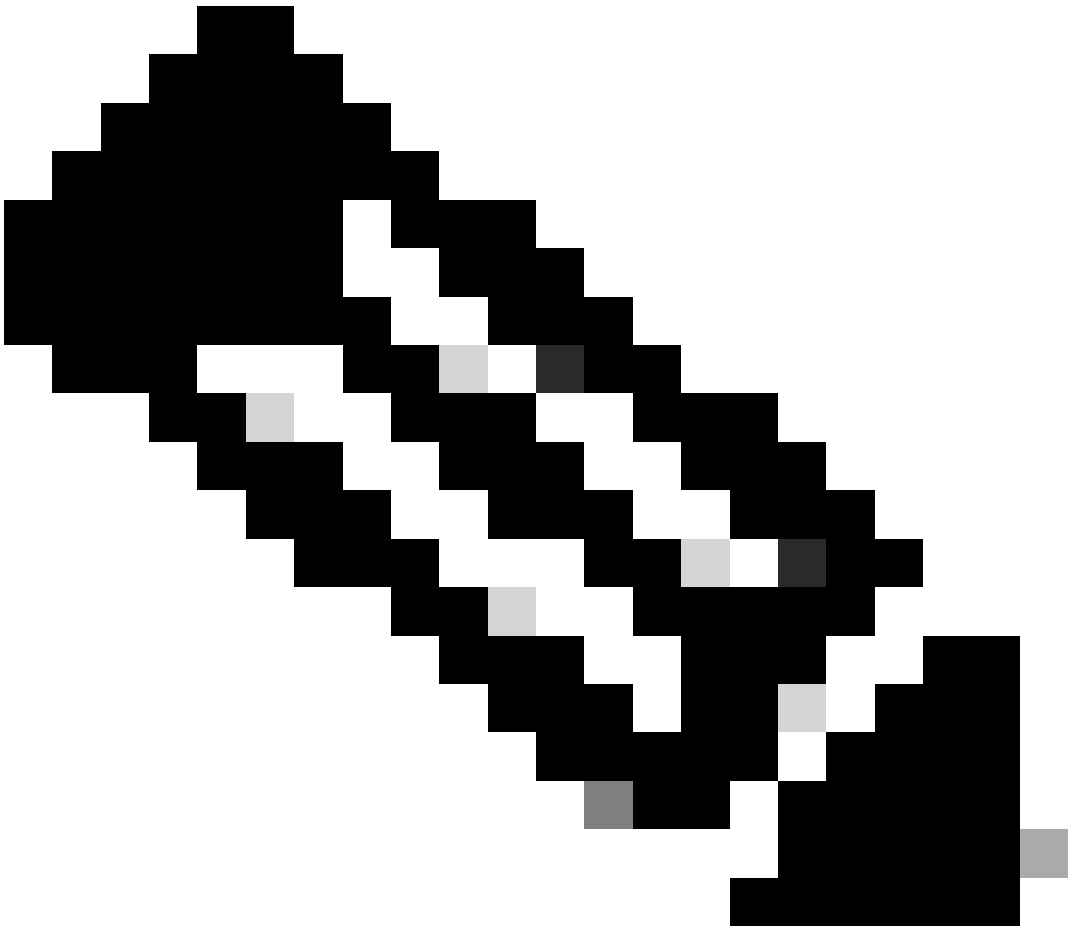
Quando um host (geralmente um PC) inicia uma sessão TCP com um servidor, ele negocia o tamanho do segmento IP usando o campo de opção MSS no pacote TCP SYN. A configuração de MTU no host determina o valor do campo MSS. O valor de MTU padrão para uma placa de rede do PC é 1500 bytes com um valor de TCP MSS de 1460 (cabeçalho IP de 1500 bytes - cabeçalho TCP de 20 bytes - cabeçalho TCP de 20 bytes).

O padrão PPP over Ethernet (PPPoE) suporta um MTU de apenas 1492 bytes.

A disparidade entre o host e o tamanho de MTU de PPPoE pode fazer com que o roteador entre o host e o servidor descarte pacotes de 1.500 bytes e encerre sessões TCP na rede PPPoE. Mesmo que o MTU do caminho (que detecta o MTU correto no caminho) esteja habilitado no host, as sessões podem ser canceladas porque os administradores do sistema às vezes desabilitam as mensagens de erro do Internet Control Message Protocol (ICMP) que devem ser retransmitidas

do host para que o MTU do caminho funcione.

O comando ip tcp adjust-mss ajuda a evitar que as sessões TCP sejam descartadas, ajustando o valor MSS dos pacotes TCP SYN. O comando ip tcp adjust-mss é efetivo apenas para conexões TCP que passam pelo roteador. Na maioria dos casos, o valor ideal para o argumento max-segment-size do comando ip tcp adjust-mss é 1452 bytes.
Esse valor mais o cabeçalho IP de 20 bytes, o cabeçalho TCP de 20 bytes e o cabeçalho PPPoE de 8 bytes somam um pacote de 1.500 bytes que corresponde ao tamanho de MTU do link Ethernet.



Observação: o tráfego baseado em ajuste do TCP MSS é comutado por software nos Switches Catalyst 9K. Este documento explica cenários que supõem que o tráfego baseado em ajuste TCP MSS é comutado por software. Consulte o Guia de configuração para confirmar se um software de HW/SW específico comuta o tráfego baseado em ajuste TCP MSS.

Comportamento

Como mencionado anteriormente, o tráfego baseado em ajuste do TCP MSS é sempre comutado por software.
Isso significa que se você tentar executar o ajuste TCP, o Switch enviará o tráfego TCP para a CPU para a modificação do MSS.

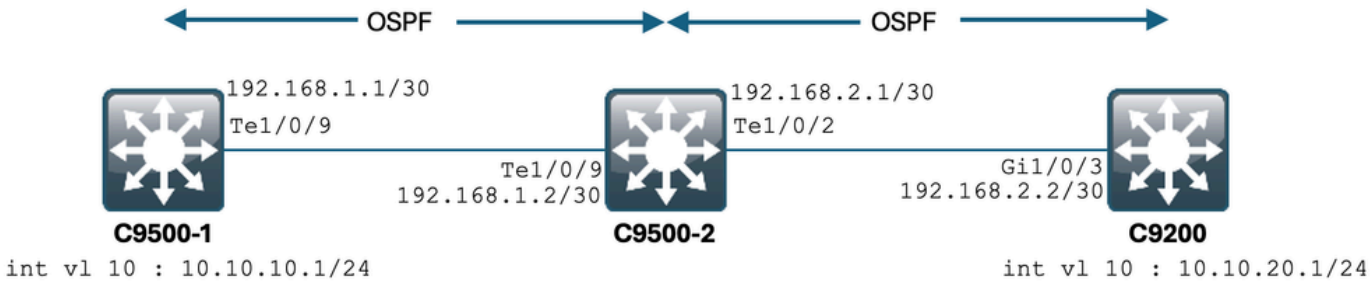
Por exemplo, se você modificar o valor TCP MSS em uma interface, todo o tráfego TCP que está sendo recebido nessa interface será direcionado para a CPU.
Em seguida, a CPU altera o valor do MSS e envia o tráfego para a interface necessária para onde o pacote TCP foi encaminhado.

Por esse motivo, se houver uma quantidade enorme de tráfego TCP com ajuste de MSS, isso sobrecarregará a fila da CPU.
Quando uma fila de CPU é sobrecarregada, o COPP (Control Plane Policer - Política de plano de controle) policia esse tráfego e descarta pacotes para manter a taxa do Queue Policer. Isso faz com que os pacotes TCP sejam descartados.

Assim, problemas como a lentidão da transferência de arquivos, as criações de sessão SSH e a lentidão do aplicativo Citrix (se estiver usando TCP) são vistos.

Um exemplo real de como isso acontece é mostrado aqui.

Topologia



Cenário

Você vai para o SSH no C9200 a partir do C9500-1.
SSH usando a VLAN 10 (10.10.10.1) do C9500-1 como a Origem.
O destino do SSH é a VLAN 20 (10.10.20.1/24) do C9200.
O SSH é baseado em TCP, portanto qualquer lentidão no TCP também afeta a criação dessa sessão SSH.

Há um Switch L3 de trânsito (C9500-2) entre C9500-1 e C9200.
Há dois links L3 de trânsito/30, um entre C9500-1 e C9500-2 e um entre C9500-2 e C9200.
O OSPF é usado para acessibilidade em todos os três Switches, e todas as/30 sub-redes e SVIs são anunciadas no OSPF.
Todos os IPs mostrados anteriormente podem ser acessados entre si.

No C9500-2 Te1/0/9, a modificação do valor TCP MSS é feita.
Quando o SSH do C9500-1 é iniciado, ocorre um handshake triplo do TCP.
O pacote SYN atinge o C9500-2 Te1/0/9 (Ingress), onde o ajuste TCP MSS é executado.

Configuração e comportamento iniciais

Uma captura EPC no C9500-2 Te1/0/9 (ambas as direções) foi feita e SSH do C9500-1 para o C9200 foi iniciado.

Esta é a configuração do EPC:

```
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#
```

Iniciando o EPC:

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

Iniciando o SSH de C9500-1 a C9200:

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

Interrupção do 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#
```

E aqui estão os pacotes capturados pelo EPC:

```
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 ~
```

Você pode ver o handshake TCP acontecendo no pacote número 1,2,3.
O pacote No.1 é o pacote SYN.
Você pode ver que ele vem com um valor MSS de 536.
O pacote SYN, ACK (Pacote No.2) também é visto vindo do C9200 com um valor de MSS de 536.
Aqui, o valor de MSS permanece intacto e não é alterado pelo Switch.

Comportamento após ajuste de TCP MSS

Aqui está a configuração de ajuste TCP MSS no 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.
```

Agora, faça uma captura EPC no C9500-2 Te1/0/9 (ambas as direções) e inicie o SSH do C9500-1 para o C9200.

Esta é a configuração do EPC:

```
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#
```

Inicie a captura, SSH de C9500-1 a C9200, e interrompa a captura.

Estes são os pacotes capturados pela CPU:

```
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 ~
```

Você pode ver o handshake TCP acontecendo nos pacotes número 2,3,4.
O pacote nº2 é o pacote SYN.
Você pode ver que ele vem com um valor MSS de 536.
Mas o pacote SYN, ACK (Pacote No.3) é visto vindo do C9200 com um valor de MSS de 512.
Isso ocorre porque quando o pacote SYN alcança o C9500-2 Te1/0/9, ele é enviado para a CPU

do C9500-2 para modificação do TCP MSS de 536 para 512.
A CPU do C9500-2 altera o MSS para 512 e envia o pacote SYN de Te1/0/2 para o C9200.
Em seguida, todas as transações TCP a seguir usam o mesmo valor de MSS modificado.

Agora, vamos nos aprofundar em como o pacote SYN atravessa o Switch e a alteração de MSS acontece.
Quando esse pacote SYN chega à interface do C9500-2, ele é enviado à CPU para modificação do MSS.
Primeiro, ele passa pelo FED (onde é possível capturá-lo) e, em seguida, vai para a CPU (onde também é possível capturá-lo).

Primeiro, vamos fazer uma captura FED Punt no C9500-2.

Esta é a configuração de captura de punt FED:

```
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
```

Iniciando a captura de punt FED:

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

Iniciando o SSH de C9500-1 a C9200:

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

Parando a captura de punt FED:

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

E aqui estão os pacotes capturados pelo punt FED:

```
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
```

Você pode ver que o Pacote No. 2 é o pacote TCP SYN de 10.10.10.1 a 10.10.20.1, vindo de Te1/0/9.
O "q-no" é importante observar aqui. Você pode ver que ele escolhe a Fila Nº 14 para ir do FED para a CPU.

Aqui você pode ver todas as 32 filas presentes para que o tráfego se mova do FED para a CPU:

```
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
```

Como você pode ver a sobrecarga, a fila nº 14 é a fila de 'encaminhamento de software'. Nesse caso, essa fila é usada pelo tráfego TCP para ser direcionada para a CPU.

Agora, vamos capturar uma CPU (plano de controle) no C9500-2.

Esta é a configuração de captura da CPU:

```
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#
```

Inicie a captura, SSH de C9500-1 a C9200, e pare a captura.

Estes são os pacotes capturados pela CPU:

```
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 ~
```

O pacote nº 12 é o pacote TCP SYN que entra na CPU (punt), com o valor padrão de MSS de 536.
O pacote nº 13 é o pacote TCP SYN enviado pela CPU (inserção), após a modificação do valor MSS para 512.

Você também pode fazer uma rápida depuração da CPU para ver essa alteração acontecendo.

Aqui está a configuração de depuração da CPU:

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

Iniciando o SSH de C9500-1 a C9200:

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

Parando a depuração da CPU:

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

Examinando os logs das depurações:

```
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#
```

Você pode ver o overhead de que o valor original do MSS de 536 está sendo ajustado para 512.

Finalmente, você pode fazer uma captura EPC em C9200 Gi1/0/3 para confirmar se o pacote TCP SYN vem de fato com um MSS de 512.

Esta é a configuração do EPC:

```
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#
```

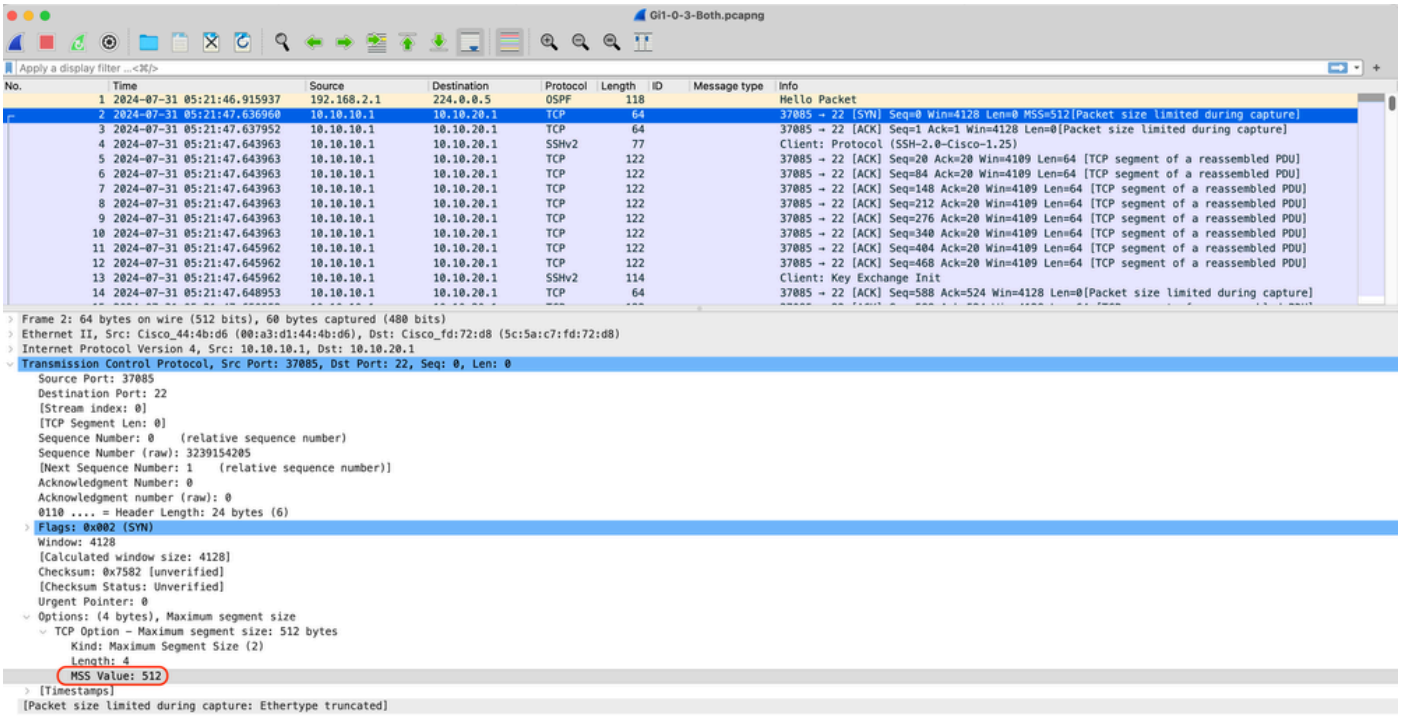
Inicie a captura, SSH de C9500-1 a C9200, e pare a captura.

Estes são os pacotes capturados pela CPU:

```
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 ~
```

No C9200, você não pode ver os detalhes do pacote como no Wireshark, apenas os detalhes breves e hexadecimais estão disponíveis.
Assim, você pode exportar os pacotes anteriores para um arquivo pcap na flash.
C9200#mon cap mycap export flash:Gi1-0-3-Both.pcapng
Exportado com sucesso

Em seguida, você pode copiar esse arquivo via TFTP para o PC local e abrir o arquivo no Wireshark.
Aqui está a captura Wireshark.



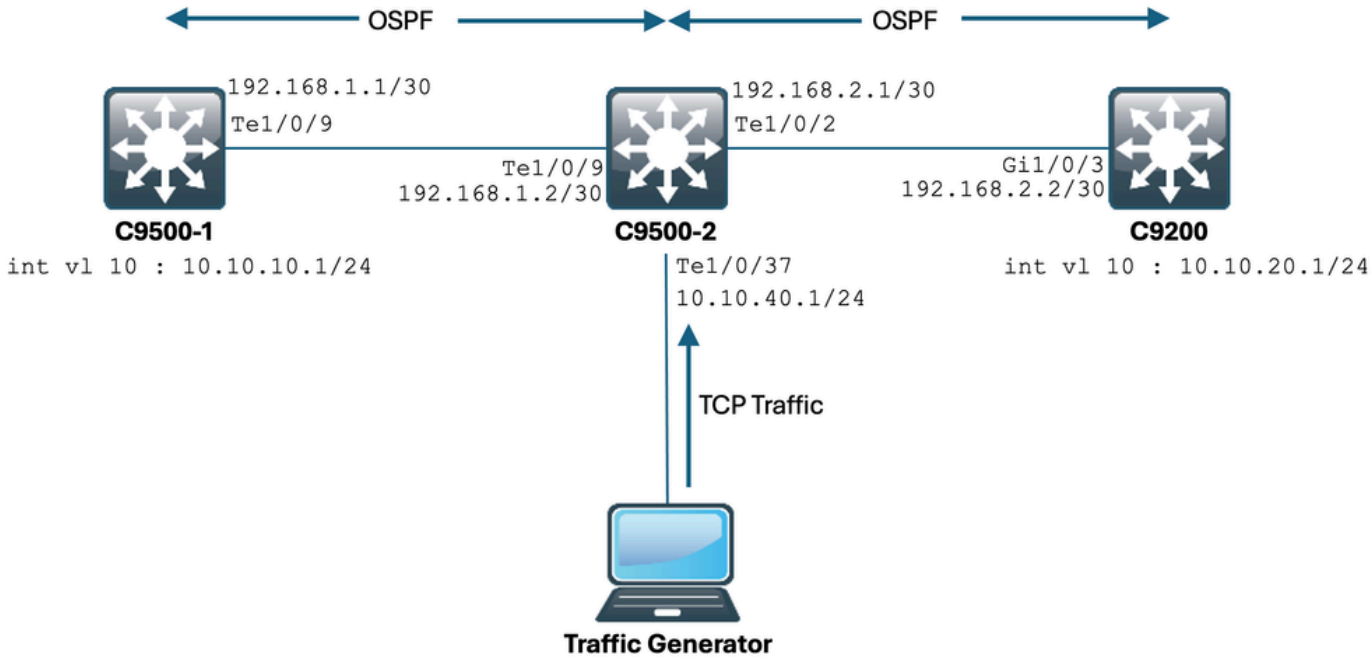
Você pode ver que o valor TCP MSS do pacote SYN é 512.

Ajuste TCP MSS causando lentidão durante uma quantidade enorme de tráfego TCP

Agora, vamos supor que uma rede tenha vários dispositivos usando tráfego TCP.
Por exemplo, eles podem transferir arquivos ou acessar um aplicativo baseado em TCP (como um servidor Citrix).

Você o simulou conectando um IXIA (gerador de tráfego) ao C9500-2 Te1/0/37, enviando pacotes TCP SYN a uma taxa alta.
Esse dispositivo IXIA atua como um segmento de rede, onde vários usuários estão usando aplicativos baseados em TCP.
Você configurou a CLI ip tcp adjust-mss em Te1/0/37.
Isso faz com que todo o tráfego TCP que está sendo recebido em Te1/0/37 seja direcionado para a CPU do C9500-2.
Isso, por sua vez, sufoca a fila "Encaminhamento de software" do agente COPP do C9500-2, como mencionado anteriormente no documento.
Como consequência, o estabelecimento da sessão SSH de C9500-1 a C9200 é afetado.
A sessão SSH não forma e atinge o tempo limite ou é estabelecida após um atraso.

Esta é a aparência da topologia:



Vamos ver isso em ação.

Esta é a configuração de C9500-2 Te1/0/37:

```
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
```

Agora você começa a enviar tráfego enorme do IXIA para a interface Te1/0/37.
Vejamos a taxa de tráfego de entrada:

```
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
```

Vamos tentar usar SSH do C9500-1 para o C9200 agora:

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

Você pode ver claramente que o C9500-1 não conseguiu fazer SSH no C9200.
Isso ocorre porque o pacote TCP SYN enviado pelo C9500-1 estava sendo descartado pela fila de 'encaminhamento de Sw', que está sendo bombardeada com tráfego de Te1/0/37.

Vamos dar uma olhada na fila:

```
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
```

Vamos coletar a saída várias vezes para garantir que a contagem descartada aumente durante o problema:

```
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#
```

Como você pode ver, a contagem de descartes está aumentando e o tráfego SSH (pacote TCP SYN) está sendo descartado aqui.

Agora, se você não souber através de qual interface/SVI está recebendo essa entrada de tráfego, terá um comando específico para ajudar.

```
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#
```

O comando show platform software fed switch ative punt rates interfaces nos fornece a lista de interfaces que são responsáveis por receber uma grande quantidade de tráfego que está sendo enviado para a CPU.
Você pode ver claramente Te1/0/37 aqui, que é a interface através da qual você está obtendo o tráfego TCP.

Agora, se você quiser ver a quantidade de tráfego que atinge todas as filas do vigilante COPP (que está sendo recebida na interface anterior), você pode usar:
show platform software fed switch ative punt rates interfaces <IF_ID da saída acima>

Vamos dar uma olhada:

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

Coletando a saída várias vezes em intervalos muito curtos:

```
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#
```

Isso mostra claramente que a fila de encaminhamento de software está bloqueada.
Uma vez que você remova o **ip tcp adjust-mss** comando do Te1/0/37, ou se você parar esse tráfego TCP, o acesso SSH do C9500-1 ao C9200 imediatamente será restabelecido.

Vamos dar uma olhada na sessão SSH após desligar o C9500-2 Te1/0/37:

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

Você pode ver que o acesso SSH é restaurado novamente.
Assim, você pode correlacionar a Lentidão do TCP aqui (acesso SSH bloqueado) devido à alta quantidade de tráfego TCP na rede, com o ajuste TCP MSS.

Pontos importantes

1. Sempre que houver lentidão de TCP em sua rede, como lentidão de transferência de arquivos, acessibilidade a aplicativos relacionados a TCP e assim por diante, e você tiver o ajuste TCP MSS configurado em um Switch Catalyst, certifique-se de verificar as quedas do Vigilante COPP para verificar se há uma grande quantidade de tráfego TCP na rede ou não.
2. Se você configurou o ajuste TCP MSS em um Switch Catalyst, certifique-se de que o tráfego TCP em sua rede não exceda a taxa do Vigilante COPP, caso contrário, problemas relacionados ao TCP (lentidão, quedas de pacotes) são vistos em sua rede.

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.