Resolución de Problemas de Lentitud TCP debido al Ajuste MSS en Catalyst 9K Switches

Contenido

Introducción

Este documento describe cómo un switch Catalyst 9K realiza el ajuste TCP MSS y cómo la lentitud TCP está vinculada a esta función.

Información sobre TCP MSS Adjustment

La función de ajuste del tamaño máximo de segmento (MSS) del protocolo de control de transmisión (TCP) habilita la configuración del tamaño máximo de segmento para los paquetes transitorios que atraviesan un router, específicamente los segmentos TCP con el bit SYN configurado. El`ip tcp adjust-mss`comando se utiliza en el modo de configuración de la interfaz para especificar el valor MSS en el router intermedio de los paquetes SYN para evitar el truncamiento.

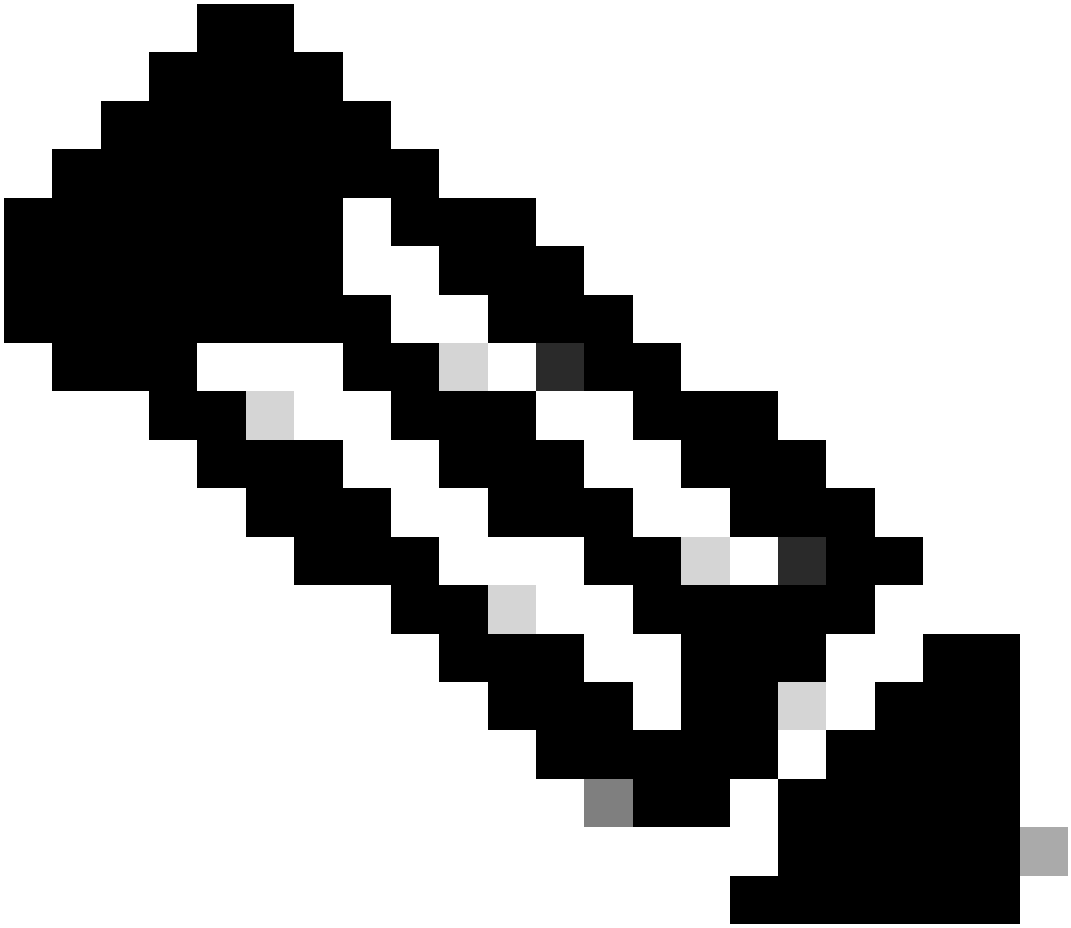
Cuando un host (normalmente un PC) inicia una sesión TCP con un servidor, negocia el tamaño del segmento IP mediante el campo de opción MSS en el paquete SYN TCP. La configuración de MTU en el host determina el valor del campo MSS. El valor predeterminado de MTU para una NIC del PC es de 1500 bytes con un valor de MSS de TCP de 1460 (encabezado IP de 20 bytes - encabezado TCP de 1500 bytes - encabezado TCP de 20 bytes).

El estándar PPP sobre Ethernet (PPPoE) admite una MTU de sólo 1492 bytes.

La disparidad entre el host y el tamaño de MTU PPPoE puede hacer que el router entre el host y el servidor descarte paquetes de 1500 bytes y termine las sesiones TCP a través de la red PPPoE.
Incluso si la MTU de trayectoria (que detecta la MTU correcta a través de la trayectoria) está habilitada en el host, las sesiones se pueden descartar porque los administradores del sistema a

veces inhabilitan los mensajes de error del Protocolo de mensajes de control de Internet (ICMP) que se deben retransmitir desde el host para que la MTU de trayectoria funcione.

El comando ip tcp adjust-mss ayuda a evitar que las sesiones TCP se pierdan mediante el ajuste del valor MSS de los paquetes SYN TCP. El comando ip tcp adjust-mss sólo es efectivo para las conexiones TCP que pasan a través del router. En la mayoría de los casos, el valor óptimo para el argumento max-segment-size del comando ip tcp adjust-mss es de 1452 bytes.
Este valor más el encabezado IP de 20 bytes, el encabezado TCP de 20 bytes y el encabezado PPPoE de 8 bytes se suman a un paquete de 1500 bytes que coincide con el tamaño de MTU para el link Ethernet.



Nota: El tráfico basado en el ajuste de TCP MSS es conmutado por software en switches Catalyst 9K. Este documento explica los escenarios que asumen que el tráfico basado en el ajuste MSS de TCP es conmutado por software. Consulte la Guía de configuración para confirmar si un software HW/SW específico conmuta el tráfico basado en el ajuste de MSS de TCP.

Comportamiento

Como se mencionó anteriormente, el tráfico basado en el ajuste de TCP MSS siempre se conmuta por software.
Esto significa que si intenta realizar el ajuste de TCP, el switch envía el tráfico TCP a la CPU para la modificación de MSS.

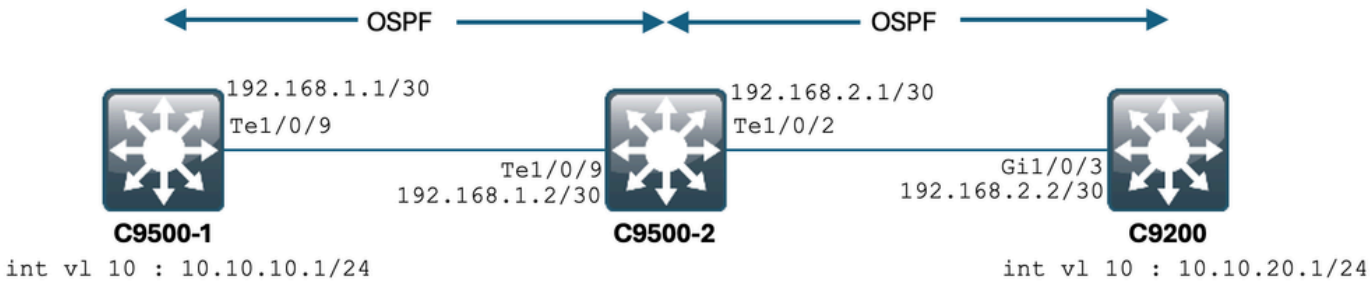
Por ejemplo, si modifica el valor TCP MSS en una interfaz, todo el tráfico TCP que se recibe en esa interfaz se envía a la CPU.
La CPU luego cambia el valor de MSS y envía el tráfico a la interfaz requerida a donde se dirigía el paquete TCP.

Debido a esta razón, si hay una enorme cantidad de tráfico TCP con ajuste MSS, esto sobrecarga la cola de la CPU.
Cuando se sobrecarga una cola de CPU, el Controlador de políticas del plano de control (COPP) controla el tráfico y descarta paquetes para mantener la velocidad del Controlador de políticas de cola. Esto hace que se descarten los paquetes TCP.

Por lo tanto, se observan problemas como la lentitud de la transferencia de archivos, las creaciones de sesiones SSH y la lentitud de las aplicaciones Citrix (si se utiliza TCP).

Aquí se muestra un ejemplo real de cómo sucede esto.

Topología



Situación

Va a SSH en el C9200 desde el C9500-1.
SSH que utiliza VLAN 10 (10.10.10.1) de C9500-1 como origen.
El destino del SSH es la VLAN 20 de C9200 (10.10.20.1/24).
SSH está basado en TCP, por lo que cualquier lentitud en TCP también afecta a la creación de esta sesión SSH.

Hay un switch L3 de tránsito (C9500-2) entre C9500-1 y C9200.
Hay dos links de tránsito /30 L3, uno entre C9500-1 y C9500-2, y otro entre C9500-2 y C9200.

OSPF se utiliza para la accesibilidad en los tres switches, y todas las subredes IP/30 y SVI se anuncian en OSPF.
Todas las IP mostradas anteriormente son accesibles entre ellas.

En C9500-2 Te1/0/9, se realiza la modificación del valor TCP MSS.
Cuando se inicia el SSH del C9500-1, se produce un intercambio de señales TCP de 3 vías.
El paquete SYN llega al C9500-2 Te1/0/9 (Entrada), donde se realiza el ajuste de MSS de TCP.

Configuración y comportamiento inicial

Se tomó una captura EPC en C9500-2 Te1/0/9 (ambas direcciones) y se inició SSH de C9500-1 a C9200.

Esta es la configuración de 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#
```

Inicio del EPC:

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

Inicio de SSH de C9500-1 a C9200:

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

Detención del 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#
```

Estos son los paquetes capturados por 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 ~
```

Puede ver el intercambio de señales TCP en el paquete número 1, 2, 3.
El paquete nº 1 es el paquete SYN.
Puede ver que viene con un valor MSS de 536.
El paquete SYN, ACK (paquete n.º 2) también se ve procedente del C9200 con un valor MSS de 536.
Aquí, el valor MSS permanece intacto y no es cambiado por el switch.

Comportamiento después del Ajuste TCP MSS

Esta es la configuración de ajuste TCP MSS en 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.
```

Ahora, tome una captura EPC en C9500-2 Te1/0/9 (ambas direcciones), e inicie SSH desde C9500-1 a C9200.

Esta es la configuración de 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 la captura, SSH de C9500-1 a C9200, y detenga la captura.

Estos son los paquetes capturados por la 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 ~
```

Puede ver el intercambio de señales TCP en los paquetes número 2, 3, 4.
El paquete nº 2 es el paquete SYN.
Puede ver que viene con un valor MSS de 536.
Sin embargo, el paquete SYN, ACK (paquete n.º 3) se ve procedente del C9200 con un valor MSS de 512.
Esto se debe a que cuando el paquete SYN alcanza el C9500-2 Te1/0/9, se envía a la CPU del C9500-2 para la modificación de TCP MSS de 536 a 512.
La CPU del C9500-2 cambia el MSS a 512 y envía el paquete SYN fuera de Te1/0/2 hacia C9200.
Entonces todas las transacciones TCP siguientes utilizan el mismo valor MSS modificado.

Ahora vamos a profundizar en cómo el paquete SYN atraviesa el switch y se produce el cambio de MSS.
Una vez que este paquete SYN alcanza la interfaz del C9500-2, se envía a la CPU para la modificación de MSS.
Primero pasa a través de la FED (donde se puede capturar), y luego va a la CPU (donde se puede capturar también).

Primero hagamos una captura de FED Punt en C9500-2.

Esta es la configuración de captura de punt de 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
```

Comenzando la captura de punt FED:

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

Inicio de SSH de C9500-1 a C9200:

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

Detención de la captura de punt de la FED:

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

Y aquí están los paquetes capturados por la 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
```

Puede ver que el Paquete N.º 2 es el paquete SYN TCP de 10.10.10.1 a 10.10.20.1, que viene desde Te1/0/9.
El "q-no" es importante tener en cuenta aquí. Puede ver que elige la Cola No. 14 para ir de la FED a la CPU.

Aquí puede ver las 32 colas presentes para que el tráfico se mueva de la FED hacia la 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 puede ver, la cola nº 14 es la cola de "reenvío de SW".
En este caso, esta cola es utilizada por el tráfico TCP para ser impulsada a la CPU.

Ahora, tomemos una captura de CPU (plano de control) en C9500-2.

Esta es la configuración de captura de 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#
```

Inicia la captura, SSH de C9500-1 a C9200, y detiene la captura.

Estos son los paquetes capturados por la 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 ~
```

El paquete nº 12 es el paquete SYN TCP que entra en la CPU (punt), con el valor MSS predeterminado de 536.
El Paquete Nº 13 es el paquete TCP SYN enviado por la CPU (inyección), después de modificar el valor MSS a 512.

También puede realizar una depuración rápida de la CPU para ver cómo se produce este cambio.

Esta es la configuración de depuración de la CPU:

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

Inicio de SSH de C9500-1 a C9200:

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

Detención de la depuración de la CPU:

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

Viendo los registros para las depuraciones:

```
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#
```

Puede ver que el valor MSS original de 536 se está ajustando a 512.

Finalmente, puede tomar una captura EPC en C9200 Gi1/0/3 para confirmar que el paquete TCP SYN efectivamente viene con un MSS de 512.

Esta es la configuración de 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#
```

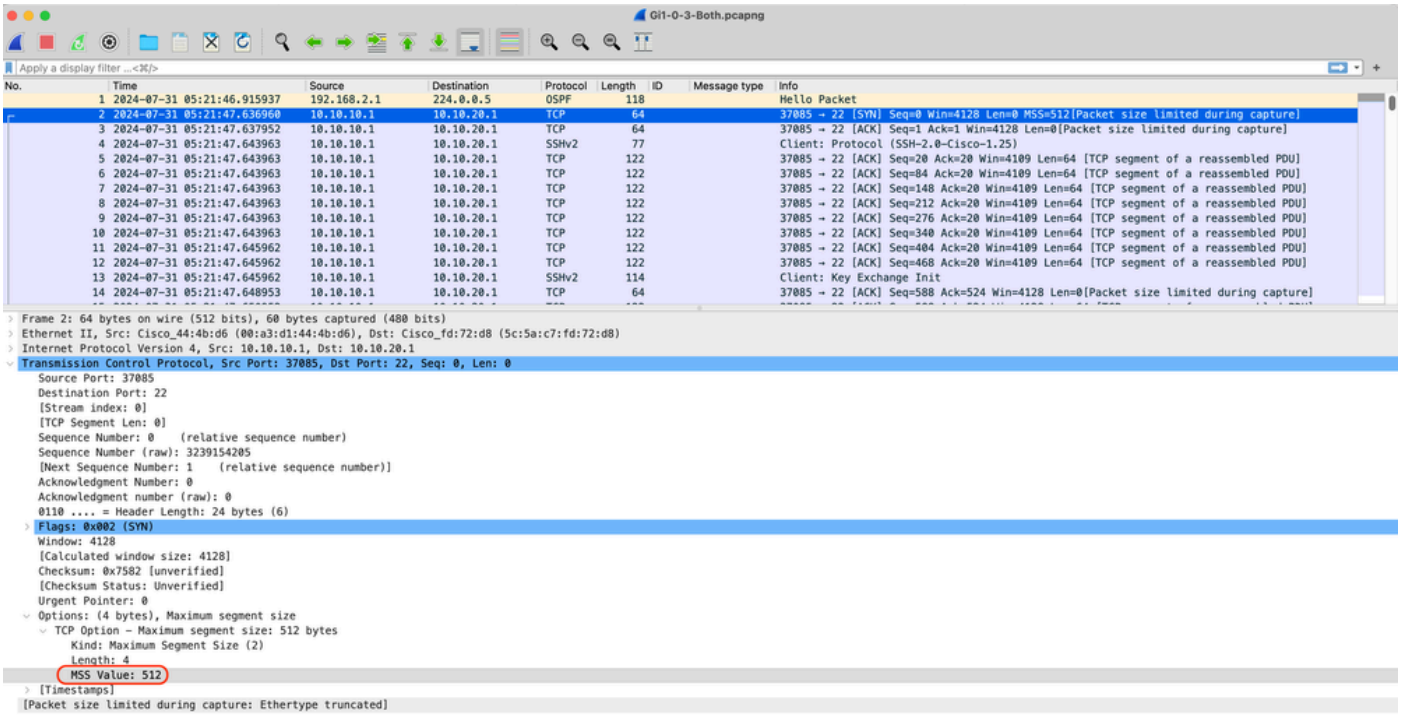
Inicia la captura, SSH de C9500-1 a C9200, y detiene la captura.

Estos son los paquetes capturados por la 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 ~
```

En C9200, no puede ver los detalles del paquete como en Wireshark, solo están disponibles los detalles breves y hexadecimales.
Por lo tanto, puede exportar los paquetes anteriores a un archivo pcap en la memoria flash.
C9200#mon cap mycap export flash:Gi1-0-3-Both.pcapng
Exportación correcta

Luego puede copiar este archivo a través de TFTP en su PC local y abrir el archivo en Wireshark.
Esta es la captura de Wireshark.



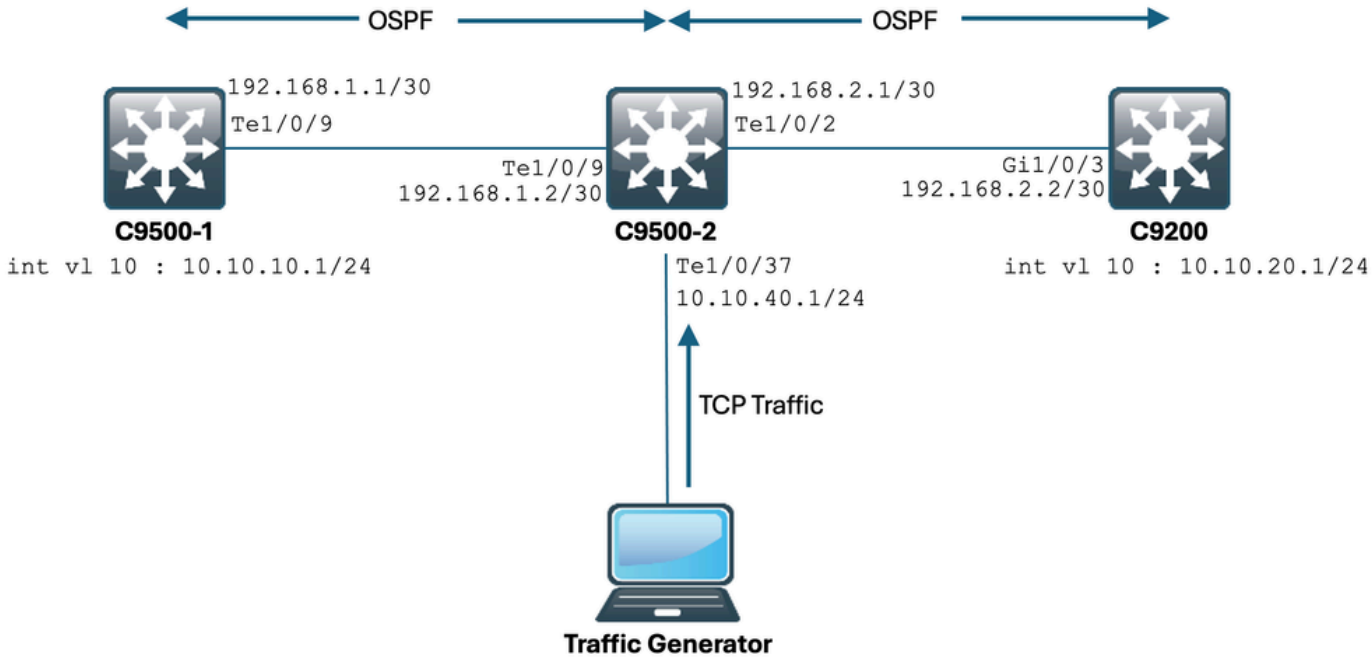
Puede ver que el valor TCP MSS del paquete SYN es 512.

Ajuste de MSS de TCP que causa lentitud durante una cantidad enorme de tráfico TCP

Supongamos que una red tiene varios dispositivos que utilizan tráfico TCP.
Por ejemplo, pueden ser la transferencia de archivos o el acceso a una aplicación basada en TCP (como un servidor Citrix).

Ha simulado esto conectando un IXIA (generador de tráfico) a C9500-2 Te1/0/37, enviando paquetes SYN TCP a una velocidad alta.
Este dispositivo IXIA actúa como un segmento de red en el que varios usuarios utilizan aplicaciones basadas en TCP.
Ha configurado ip tcp adjust-mss CLI en Te1/0/37.
Esto hace que todo el tráfico TCP que se recibe en Te1/0/37 se dirija a la CPU de C9500-2.
Esto, a su vez, estrangula la cola de "reenvío de SW" del regulador COPP del C9500-2, como se mencionó anteriormente en el documento.
Como consecuencia, se ve afectado el establecimiento de sesión SSH desde C9500-1 a C9200.
La sesión SSH no se forma y se agota el tiempo de espera, o se establece después de un retraso.

Este es el aspecto de la topología:



Veamos esto en acción.

Esta es la configuración 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
```

Ahora comienza a enviar un tráfico enorme desde IXIA a la interfaz Te1/0/37.
Echemos un vistazo a la velocidad del tráfico entrante:

```
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
```

Ahora intentemos usar SSH de C9500-1 a C9200:

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

Puede ver claramente que el C9500-1 no pudo introducir SSH en el C9200.
Esto se debe a que el paquete TCP SYN que envía el C9500-1 fue descartado por la cola de "reenvío de SW", que está siendo bombardeada con tráfico de Te1/0/37.

Echemos un vistazo a la cola:

```
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
```

Recolectemos el resultado varias veces para asegurarnos de que el conteo descartado está aumentando durante el 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 puede ver, el conteo descartado está aumentando, y el tráfico SSH (paquete TCP SYN) se está descartando aquí.

Ahora, si no sabe a través de qué interfaz/SVI está recibiendo este flujo de tráfico, tiene un comando específico para ayudarle.

```
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#
```

El comando show platform software fed switch active punt rates interfaces nos da la lista de interfaces que son responsables de recibir una enorme cantidad de tráfico que se envía a la CPU. Aquí puede ver claramente Te1/0/37, que es la interfaz a través de la cual obtiene el tráfico TCP.

Ahora, si desea ver la cantidad de tráfico que llega a todas las colas del Policer COPP (que se recibe en la interfaz anterior), puede utilizar:
show platform software fed switch active punt rates interfaces <IF_ID from the above output>

Vamos a echar un vistazo:

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

Recolección de la salida varias veces en intervalos muy cortos:

```
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#
```

Esto muestra claramente que la cola de reenvío de SW está bloqueada.
Una vez que quite el **ip tcp adjust-mss** comando del Te1/0/37, o si detiene este tráfico TCP, el acceso SSH de C9500-1 a C9200 se restablece inmediatamente.

Echemos un vistazo a la sesión SSH después de apagar C9500-2 Te1/0/37:

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

Puede ver que el acceso SSH se restaura nuevamente.
Por lo tanto, puede correlacionar la Lentitud TCP aquí (acceso SSH bloqueado) debido a la alta cantidad de tráfico TCP en la red, con el ajuste TCP MSS.

Puntos importantes

1. Siempre que tenga lentitud TCP en su red, como lentitud en la transferencia de archivos, accesibilidad a aplicaciones relacionadas con TCP, etc., y tenga configurado el ajuste TCP MSS en un switch Catalyst, asegúrese de verificar las caídas del regulador COPP para verificar si hay una gran cantidad de tráfico TCP en la red o no.
2. Si ha configurado el ajuste de MSS de TCP en un switch Catalyst, asegúrese de que el

tráfico TCP de su red no se suscriba en exceso a la velocidad del regulador de COPP; de lo contrario, se observarán en su red problemas relacionados con TCP (lentitud, caídas de paquetes).

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).