

# Catalyst 9KスイッチでのMSS調整によるTCP速度低下の問題のトラブルシューティング

## 内容

---

[はじめに](#)

[TCP MSSの調整に関する情報](#)

[動作](#)

[トポロジ](#)

[シナリオ](#)

[初期設定と動作](#)

[TCP MSS調整後の動作](#)

[膨大な量のTCPトラフィックの遅延を引き起こすTCP MSSの調整](#)

[重要なポイント](#)

---

## はじめに

このドキュメントでは、Catalyst 9KスイッチでTCP MSSの調整がどのように行われるか、およびこの機能にTCPの速度低下がどのようにリンクされるかについて説明します。

## TCP MSSの調整に関する情報

Transmission Control Protocol ( TCP ; 伝送制御プロトコル ) のMaximum Segment Size ( MSS ; 最大セグメントサイズ ) Adjustment機能を使用すると、ルータを通過する一時的なパケット、特にSYNビットが設定されたTCPセグメントの最大セグメントサイズを設定できます。ip tcp adjust-mssコマンドは、トランケートを回避するために、インターフェイスコンフィギュレーションモードでSYNパケットの中継ルータのMSS値を指定するために使用されます。

ホスト ( 通常はPC ) がサーバとのTCPセッションを開始すると、TCP SYNパケットのMSSオプションフィールドを使用してIPセグメントサイズをネゴシエートします。ホストのMTU設定によって、MSSフィールドの値が決まります。PCのNICのデフォルトのMTU値は1500バイトで、TCP MSS値は1460です ( 1500バイト - 20バイトのIPヘッダー - 20バイトのTCPヘッダー ) 。

PPP over Ethernet(PPPoE)標準では、1492バイトのMTUだけがサポートされています。

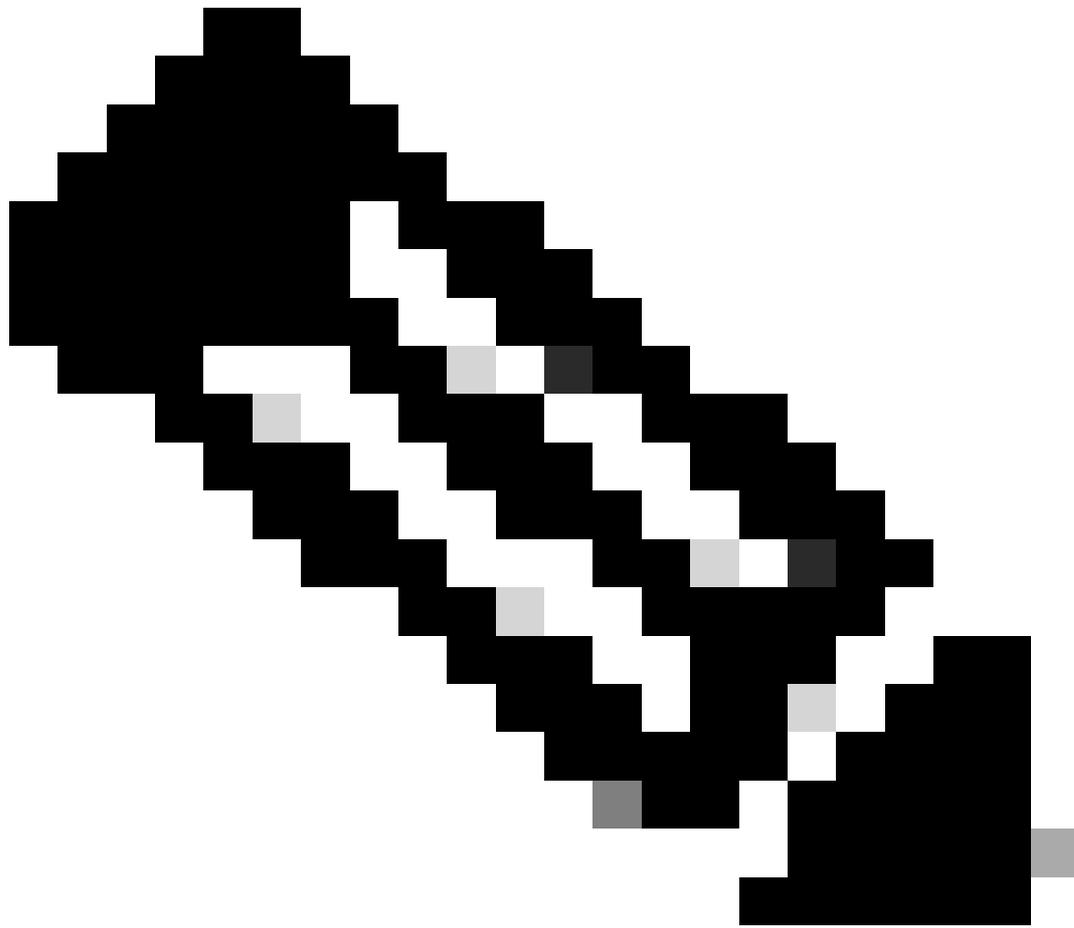
ホストとPPPoE MTUサイズが一致しない場合、ホストとサーバ間のルータは1500バイトのパケットをドロップし、PPPoEネットワーク上のTCPセッションを終了します。パスMTU ( パスを通過する正しいMTUを検出する ) がホストで有効になっている場合でも、システム管理者がInternet Control Message Protocol(ICMP)エラーメッセージを無効にすることがあります。その場合、パスMTUが機能するためにはホストからリレーする必要があります。

ip tcp adjust-mssコマンドでは、TCP SYNパケットのMSS値を調整することで、TCPセッション

の廃棄を防止できます。ip tcp adjust-mssコマンドは、ルータを通過するTCP接続に対してのみ有効です。ほとんどの場合、ip tcp adjust-mssコマンドのmax-segment-size引数の最適値は1452バイトです。

この値に、20バイトのIPヘッダー、20バイトのTCPヘッダー、および8バイトのPPPoEヘッダーを加えた値が、イーサネットリンクのMTUサイズと一致する1500バイトのパケットになります。

---



注:TCP MSS調整ベースのトラフィックは、Catalyst 9000スイッチでソフトウェアスイッチングされます。このドキュメントでは、TCP MSS調整ベースのトラフィックがソフトウェアでスイッチングされることを前提としたシナリオについて説明します。特定のハードウェア/ソフトウェアがTCP MSS調整ベースのトラフィックを切り替えているかどうかを確認するには、『[コンフィギュレーションガイド](#)』を参照してください。

---

## 動作

前述したように、TCP MSS調整ベースのトラフィックは、常にソフトウェアでスイッチングされます。

つまり、TCPの調整を試みて実行すると、スイッチはMSSの変更のためにTCPトラフィックをCPUに送信します。

たとえば、インターフェイスのTCP MSS値を変更すると、そのインターフェイスで受信されているすべてのTCPトラフィックがCPUにパントされます。

次にCPUはMSS値を変更し、そのTCPパケットが送信された必要なインターフェイスにトラフィックを送信します。

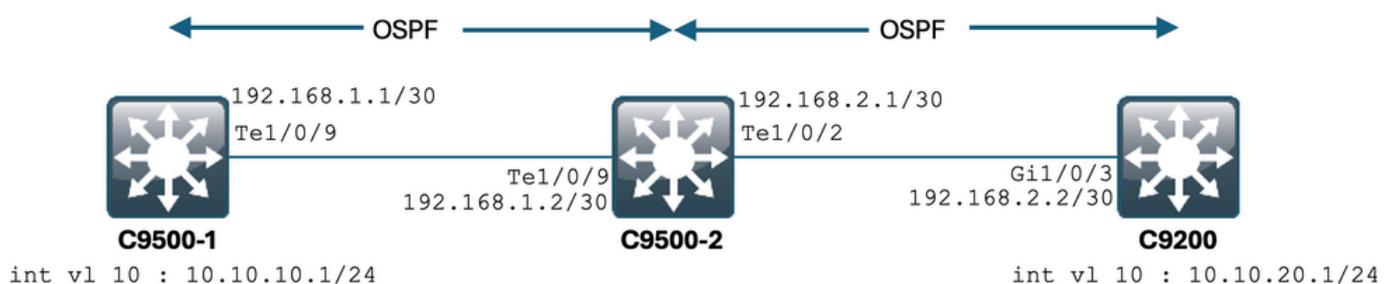
このため、MSSの調整によって大量のTCPトラフィックが発生すると、CPUキューが過負荷になります。

CPUキューが過負荷状態になると、コントロールプレーンポリサー(COPP)は、キューポリサーレートを維持するために、そのトラフィックをポリシングし、パケットをドロップします。これにより、TCPパケットがドロップされます。

したがって、ファイル転送の速度低下、SSHセッションの作成、Citrixアプリケーションの速度低下 (TCPを使用している場合) などの問題が発生します。

この動作の実例を次に示します。

## トポロジ



## シナリオ

C9500-1からC9200にSSH接続します。

C9500-1のVLAN 10(10.10.10.1)を送信元として使用してSSHします。

SSHの宛先はC9200のVLAN 20(10.10.20.1/24)です。

SSHはTCPベースであるため、TCPの速度低下もこのSSHセッションの作成に影響します。

C9500-1とC9200の間にトランジットL3スイッチ(C9500-2)があります。

2つのトランジット/30 L3リンクがあり、1つはC9500-1とC9500-2の間、もう1つはC9500-2とC9200の間にあります。

OSPFは3つすべてのスイッチ間の到達可能性に使用され、すべての/30サブネットとSVIがOSPFにアドバタイズされます。

先に示したすべてのIPは、相互に到達可能です。

C9500-2 Te1/0/9では、TCP MSS値の変更が行われます。

C9500-1からSSHが開始されると、TCP 3ウェイハンドシェイクが発生します。  
SYNパケットはC9500-2 Te1/0/9 ( 入力 ) に到達し、そこでTCP MSS調整が実行されます。

## 初期設定と動作

C9500-2 Te1/0/9 ( 両方向 ) のEPCキャプチャが取得され、C9500-1からC9200へのSSHが開始されました。

EPC設定を次に示します。

```
C9500-2#show monitor capture mycap
Status Information for Capture mycap
Target Type:
Interface: TenGigabitEthernet1/0/9, Direction: BOTH
Status : Inactive
Filter Details:
Capture all packets
Buffer Details:
Buffer Type: LINEAR (default)
Buffer Size (in MB): 80
File Details:
File not associated
Limit Details:
Number of Packets to capture: 0 (no limit)
Packet Capture duration: 0 (no limit)
Packet Size to capture: 0 (no limit)
Maximum number of packets to capture per second: 1000
Packet sampling rate: 0 (no sampling)
C9500-2#
```

EPCの起動 :

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

C9500-1からC9200へのSSHの開始 :

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

EPCの停止 :

```
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 exist till exported or cleared
Stopped capture point : mycap
C9500-2#
```

EPCでキャプチャされたパケットを次に示します。

```
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 segment]
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 segment]
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 segment]
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 segment]
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 segment]
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 ~
```

パケット番号1、2、3でTCPハンドシェイクが発生していることがわかります。

パケットNo.1はSYNパケットです。

536のMSS値が設定されていることがわかります。

SYN、ACKパケット(パケット番号2)も、MSS値536でC9200から送信されていることが確認できます。

ここでは、MSS値はそのまま残り、スイッチによって変更されることはありません。

## TCP MSS調整後の動作

C9500-2 Te1/0/9でのTCP MSSの調整の設定を次に示します。

```
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.
```

ここで、C9500-2 Te1/0/9 ( 両方向 ) のEPCキャプチャを取得し、C9500-1からC9200へのSSHを開始します。

EPC設定を次に示します。

```
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#
```

キャプチャを開始し、C9500-1からC9200にSSH接続して、キャプチャを停止します。

CPUによってキャプチャされたパケットを次に示します。

```
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 ~
```

パケット番号2、3、4でTCPハンドシェイクが発生していることがわかります。

パケットNo.2はSYNパケットです。

536のMSS値が設定されていることがわかります。

しかし、SYN、ACKパケット ( パケットNo.3 ) が512のMSS値でC9200から着信していることが確認されています。

これは、SYNパケットがC9500-2 Te1/0/9に到達すると、536から512へのTCP MSS変更のためにC9500-2のCPUに送信されるためです。

C9500-2のCPUはMSSを512に変更し、Te1/0/2からC9200に向けてSYNパケットを送信します。その後のTCPトランザクションはすべて、同じ変更後のMSS値を使用します。

次に、SYNパケットがスイッチを通過し、MSSの変更がどのように行われるかについて詳しく説明します。

このSYNパケットがC9500-2のインターフェイスに到達すると、MSSの変更のためにCPUに送信されます。

最初にFEDを通過し ( キャプチャ可能 )、次にCPUに送られます ( キャプチャ可能 )。

まず、C9500-2でのFEDパントキャプチャを取り上げます。

FEDパントキャプチャの設定を次に示します。

```
C9500-2#debug platform software fed switch 1 punt packet-capture buffer limit 16384
Punt PCAP buffer configure: one-time with buffer size 16384...done
```

FEDパントキャプチャを開始します。

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

C9500-1からC9200へのSSHの開始 :

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

FEDパントキャプチャを停止します。

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

FEDパントでキャプチャされたパケットを次に示します。

```
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], pa1: TenGigabitEthernet1/0/9 [if-id: 0x00000040]
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], pa1: TenGigabitEthernet1/0/9 [if-id: 0x00000040]
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], pa1: TenGigabitEthernet1/0/1 [if-id: 0x00000009]
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
```

パケットNo. 2は10.10.10.1から10.10.20.1へのTCP SYNパケットであり、Te1/0/9から着信していることがわかります。

ここでは、「q-no」が重要です。FEDからCPUに移動するためにキュー番号14が選択されていることがわかります。

ここでは、FEDからCPUに向かうトラフィック用に存在する32個のキューをすべて確認できます。

```
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
```

オーバーヘッドを確認できるように、キュー番号14は「Sw forwarding」キューです。  
この場合、このキューはTCPトラフィックがCPUにパントするために使用されます。

次に、C9500-2のCPU ( コントロールプレーン ) キャプチャについて説明します。

CPUキャプチャの設定を次に示します。

```
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#
```

キャプチャを開始し、C9500-1からC9200にSSH接続して、キャプチャを停止します。

CPUによってキャプチャされたパケットを次に示します。

```
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 = bg118-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 = bg118-cx-amx-b02-2.cisco
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 MSS=512
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 = 0
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 = 0
~ Output Cut ~
```

パケットNo. 12は、CPUに着信するTCP SYNパケット(punt)であり、デフォルトのMSS値は536です。

パケットNo. 13は、MSS値を512に変更した後にCPU ( 注入 ) によって送信されるTCP SYNパケットです。

また、この変更を確認するために、CPUのクイックデバッグを実行することもできます。

CPUのデバッグ設定を次に示します。

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

C9500-1からC9200へのSSHの開始 :

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

CPUデバッグの停止 :

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

ログでデバッグを確認します。

```
C9500-2#show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml disabled)
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#
```

536の元のMSS値が512に調整されていることがわかります。

最後に、TCP SYNパケットが実際に512のMSSで着信していることを確認するために、C9200 Gi1/0/3のEPCキャプチャを取得できます。

EPC設定を次に示します。

```
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#
```

キャプチャを開始し、C9500-1からC9200にSSH接続して、キャプチャを停止します。

CPUによってキャプチャされたパケットを次に示します。

```
C9200#sh mon cap mycap buff br
```

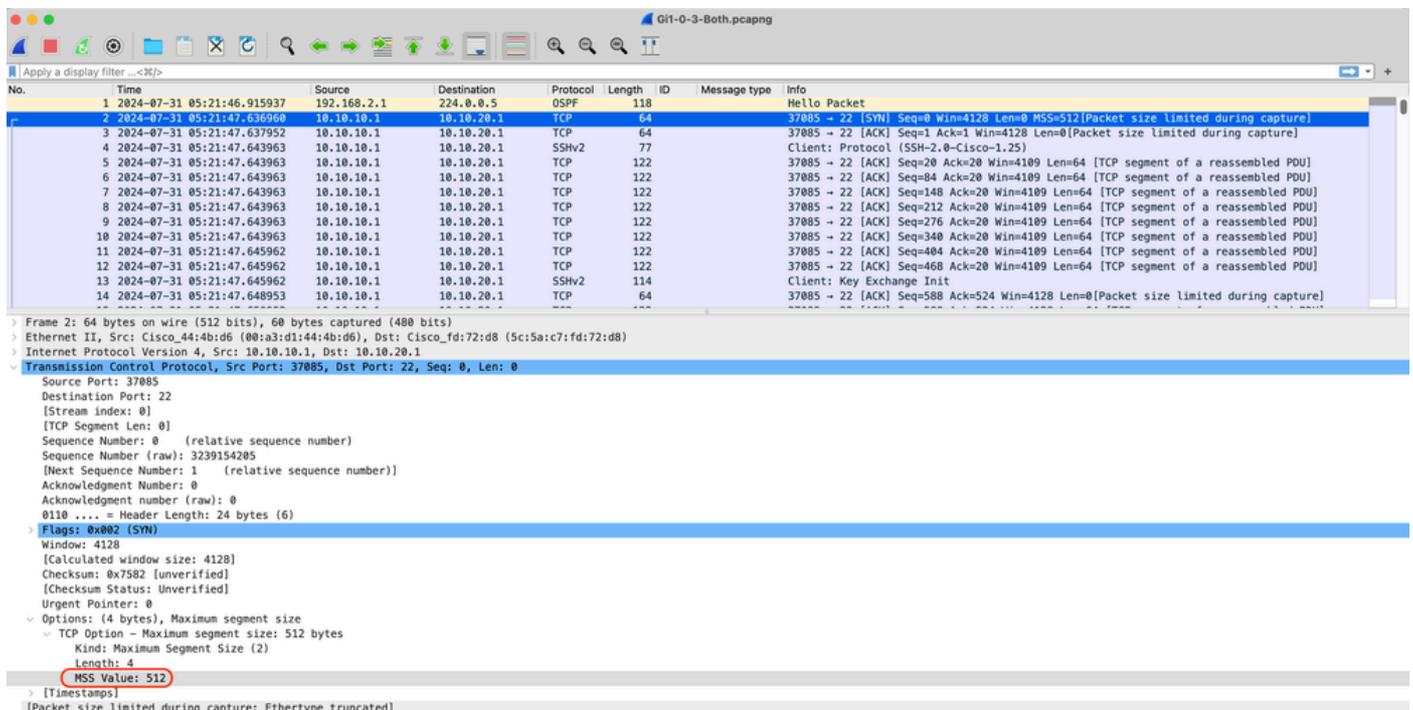
```
-----  
# size timestamp source destination dscp protocol  
-----  
0 118 0.000000 192.168.2.1 -> 224.0.0.5 48 CS6 OSPF  
1 64 0.721023 10.10.10.1 -> 10.10.20.1 48 CS6 TCP  
2 64 0.722015 10.10.10.1 -> 10.10.20.1 48 CS6 TCP  
3 77 0.728026 10.10.10.1 -> 10.10.20.1 48 CS6 TCP  
4 122 0.728026 10.10.10.1 -> 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 ~
```

C9200では、Wiresharkのようにパケットの詳細を表示することはできません。短い形式と16進形式の詳細のみが表示されます。

したがって、フラッシュ内のpcapファイルに以前のパケットをエクスポートできます。

```
C9200#mon cap mycap export flash:Gi1-0-3-Both.pcapng  
正常にエクスポート
```

次に、このファイルをTFTP経由でローカルPCにコピーし、Wiresharkで開きます。  
Wiresharkキャプチャを次に示します。



SYNパケットのTCP MSS値が512であることがわかります。

## 膨大な量のTCPトラフィックの遅延を引き起こすTCP MSSの調整

ここで、ネットワークにTCPトラフィックを使用する複数のデバイスがあると仮定します。  
たとえば、ファイルを転送したり、TCPベースのアプリケーション ( Citrix Serverなど ) にアクセスしたりできます。

シミュレーションでは、IXIA ( トラフィックジェネレータ ) をC9500-2 Te1/0/37に接続し、TCP SYNパケットを高速で送信しました。

このIXIAデバイスは、複数のユーザがTCPベースのアプリケーションを使用するネットワークセグメントとして機能します。

Te1/0/37でip tcp adjust-mss CLIを設定している。

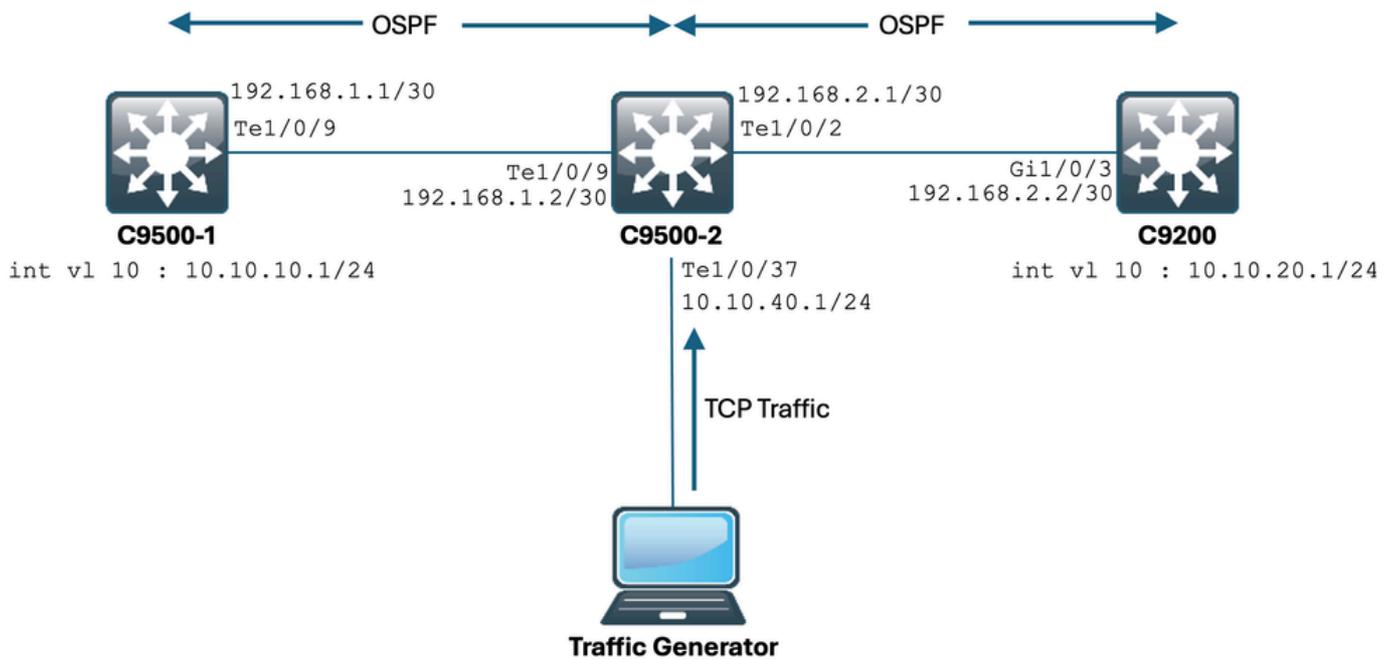
これにより、Te1/0/37で受信されるすべてのTCPトラフィックがC9500-2のCPUにパントされます。

この結果、前述したように、C9500-2のCOPPポリサーの「Sw forwarding」キューが停止します。

その結果、C9500-1からC9200へのSSHセッションの確立が影響を受けます。

SSHセッションが形成されず、タイムアウトになるか、遅延の後で確立されます。

トポロジは次のようになります。



実際に行ってみましょう。

C9500-2 Te1/0/37の設定を次に示します。

```
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
```

ここで、IXIAからTe1/0/37インターフェイスに大量のトラフィックを送信し始めます。  
着信トラフィックレートを見てみましょう。

```
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
```

ここで、C9500-1からC9200へのSSHを試行してみましょう。

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

C9500-1がC9200にSSH接続できなかったことは明らかです。  
これは、C9500-1によって送信されているTCP SYNパケットが「Sw forwarding」キューによってドロップされ、Te1/0/37からのトラフィックが大量に送られているためです。

次に、キューについて説明します。

```
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

```

問題発生時に廃棄カウントが増加していることを確認するために、出力を複数回収集してみましよう。

```

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#

```

ドロップ数が増加し、SSHトラフィック (TCP SYNパケット) はここでドロップされていることがわかります。

どのインターフェイス/SVIを経由してトラフィックの流入を受けているかがわからない場合は、特定のコマンドを使用して対処します。

```

C9500-2#show platform software fed switch active punt rates interfaces
Punt Rate on Interfaces Statistics
Packets per second averaged over 10 seconds, 1 min and 5 mins
=====
| | Recv | Recv | Recv | Drop | Drop | Drop
Interface Name | IF_ID | 10s | 1min | 5min | 10s | 1min | 5min
=====
TenGigabitEthernet1/0/37 0x00000042 1000 1000 1000 0 0 0
-----
C9500-2#

```

show platform software fed switch active punt rates interfacesコマンドを実行すると、CPUにパン  
トされている大量のトラフィックの受信を担当しているインターフェイスのリストが表示されま  
す。

Te1/0/37がTCPトラフィックを受信しているインターフェイスであることがわかります。

すべてのCOPPポリサーキュー ( 以前のインターフェイスで受信 ) にヒットするトラフィック量  
を確認するには、次のコマンドを使用します。

show platform software fed switch active punt rates interfaces <上記の出力のIF\_ID>

次の点を確認します。

```
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
-----
```

非常に短い間隔で複数回出力を収集します。

```
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#
```

これは、ソフトウェア転送キューが停止していることを明確に示しています。

Te1/0/37からip tcp adjust-mssコマンドを削除するか、このTCPトラフィックを停止すると、C9500-1からC9200へのSSHアクセスがただちに再確立されます。

C9500-2 Te1/0/37をシャットダウンした後のSSHセッションを見てみましょう。

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

SSHアクセスが再び復元されたことがわかります。

したがって、ネットワーク内の大量のTCPトラフィックによるTCPの速度低下 (SSHアクセスのブロック) を、TCP MSSの調整と関連付けることができます。

## 重要なポイント

1. ファイル転送の速度低下、TCP関連アプリケーションへのアクセシビリティなど、ネットワークにTCPの速度低下があり、CatalystスイッチでTCP MSSの調整を設定している場合は常に、COPPポリサーのドロップをチェックして、ネットワークに大量のTCPトラフィックが存在するかどうかを確認してください。
2. CatalystスイッチでTCP MSS調整を設定している場合は、ネットワークのTCPトラフィックがCOPPポリサーレートをオーバーサブスクライブしていないことを確認してください。オーバーサブスクライブしていない場合、ネットワークでTCP関連の問題 (速度低下、パケットドロップ) が発生します。

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。