

# Cisco IOS XR および Cisco IOS ルータの MTU の動作

## 内容

[概要](#)

[背景説明](#)

[設定](#)

[Cisco IOS と Cisco IOS XR ソフトウェアの比較](#)

[ルーテッド L3 インターフェイス](#)

[デフォルト MTU](#)

[非デフォルト MTU](#)

[ルーテッド L3 サブインターフェイス](#)

[L2VPN L2 インターフェイス](#)

[EVC \( ASR9000 \)](#)

[非 EVC \( XR 12000 および CRS \)](#)

[自動イーサネット インターフェイス ドライバ MTU および MRU の設定](#)

[リリース 5.1.1 よりも前のリリースからリリース 5.1.1 以降にアップグレードする際の設定の変換](#)

## 概要

このドキュメントでは、Cisco IOS® XR ルータの最大伝送ユニット ( MTU ) の動作について説明し、その動作を Cisco IOS ルータと比較します。また、イーサネット仮想コネクション ( EVC ) モデルと非 EVC モデルの両方を使用するルーテッドレイヤ 3 ( L3 ) インターフェイスとレイヤ 2 VPN ( L2VPN ) L2 インターフェイス上の MTU についても説明します。さらに、リリース 5.1.1 以降でのイーサネット インターフェイス ドライバ MTU と最大受信ユニット ( MRU ) の自動設定方法に関する重要な変更についても説明します。

## 背景説明

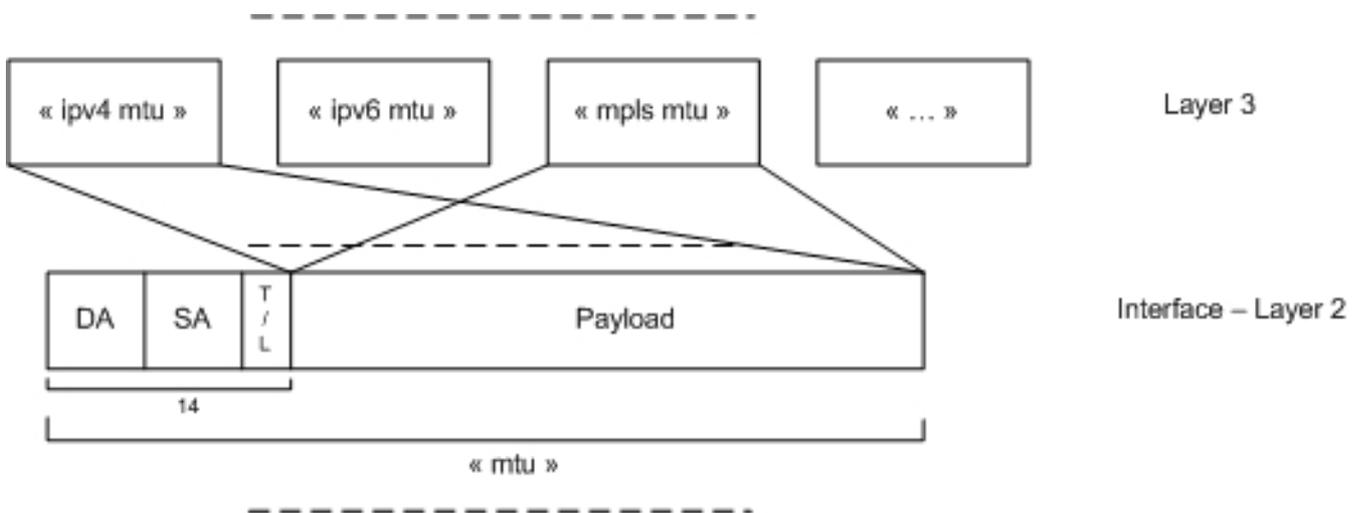
コンピュータ ネットワークでは、レイヤの通信プロトコルの MTU により、レイヤが 1 つのインターフェイスで送信できる最大プロトコル データ ユニットのサイズ ( バイト単位 ) が定義されます。各インターフェイス、レイヤ、およびプロトコルに 1 つの MTU パラメータが関連付けられます。

Cisco IOS XR ソフトウェアの MTU の特性は次のとおりです。

- MTU config および show コマンドには、L2 および L3 において、対応するレイヤのヘッダーサイズが含まれています。たとえば、L2 MTU を設定する mtu コマンドには、イーサネット インターフェイス ( dot1q なし ) の場合は 14 バイトが含まれており、ポイントツーポイント プロトコル ( PPP ) またはハイレベル データリンク コントロール ( HDLC ) の場合は 4 バイト

トが含まれています。ipv4 mtu コマンドには、IPv4 ヘッダーの 20 バイトが含まれています。

- 上位レイヤの MTU は、下位レイヤのペイロード内に収まる必要があります。たとえば、非 dot1q イーサネット インターフェイスのインターフェイス MTU がデフォルトの 1514 バイトの場合、マルチプロトコル ラベル スイッチング (MPLS) などの上位レイヤ プロトコルはそのインターフェイス上で最大 1500 バイトの MTU を持つことができます。これは、イーサネット フレーム内に 1500 バイトの MPLS フレーム (ラベルを含む) しか収めることができないことを意味します。1500 バイトの IPv4 パケットの先頭に 2 つの MPLS タグを割り当てる場合は、そのインターフェイスで 1508 バイトの MPLS MTU を設定することはできません。イーサネット インターフェイスで 1508 バイトの MPLS フレームを送信するためには、インターフェイス MTU を 1522 以上の値に増やす必要があります。そうすることで、L2 インターフェイスのペイロードが MPLS フレームを伝送するのに十分な大きさになります。



- 従来の Cisco IOS ソフトウェア (Cisco IOS XR ソフトウェアではない) では、interface mtu コマンドによって L2 ペイロード サイズが設定されますが、L2 ヘッダーは含められません。これは、interface mtu コマンドに L2 および L3 両方のオーバーヘッドを含む Cisco IOS XR ソフトウェアとは異なります。L3 MTU コマンドでは、ipv4 mtu コマンドと同様に、対象のプロトコルの最大パケット サイズ (L3 ヘッダーを含む) が設定されます。これは Cisco IOS XR ソフトウェアに似ています。
- Cisco IOS XR ソフトウェアのデフォルトのインターフェイス MTU は、1500 バイトの L3 パケットの伝送を可能にする必要があります。したがって、デフォルトの MTU は、メインのイーサネット インターフェイスの場合は 1514 バイトで、シリアル インターフェイスの場合は 1504 バイトです。

以下では、MTU の特性を説明し、Cisco IOS ソフトウェアと Cisco IOS XR ソフトウェアの動作を比較するとともに、これらのタイプのインターフェイスの例を提示します。

- ルーテッド L3 インターフェイス
- ルーテッド L3 サブインターフェイス
- L2VPN L2 インターフェイス

## 設定

注：このセクションで使用されているコマンドの詳細を調べるには、[Command Lookup Tool](#)(登録ユーザ専用)を使用してください。

注：特定のshowコマンドは、[Output Interpreter Tool](#)(登録ユーザ専用)でサポートされています。show コマンドの出力の分析を表示するには、Output Interpreter Tool を使用します。

## Cisco IOS と Cisco IOS XR ソフトウェアの比較

このセクションでは、MTU の特性に関連して Cisco IOS ソフトウェアと Cisco IOS XR ソフトウェアの動作を比較します。

Cisco IOS ソフトウェアでは、mtu コマンドと対応する show コマンドには L2 ヘッダーは含まれていません。L2 ペイロードを、L3 ヘッダーを含む L3 パケットの最大サイズに設定するには、mtu コマンドを使用します。

これは、mtu コマンドに L2 ヘッダーが含まれる (イーサネットの場合は 14 バイト、PPP/HDLC の場合は 4 バイト) Cisco IOS XR ソフトウェアとは異なります。

Cisco IOS ルータが mtu x で設定され、Cisco IOS XR ルータに接続されている場合、Cisco IOS XR ルータ上の対応するインターフェイスは、イーサネット インターフェイスの場合は mtu x+14、シリアル インターフェイスの場合は mtu x+4 で設定する必要があります。

Cisco IOSとCisco IOS XRソフトウェアでは、ipv4 mtu、ipv6 mtu、およびmpls mtuコマンドの意味が同じです。これらのコマンドは同じ値で設定する必要があります。

そのため、Cisco IOS ソフトウェアのイーサネット インターフェイスでの設定は次のようになります。

```
mtu 9012
ipv4 mtu 9000
ipv6 mtu 9000
```

Cisco IOS XR ソフトウェア ネイバーでの対応する設定は次のとおりです。

```
mtu 9026
ipv4 mtu 9000
ipv6 mtu 9000
```

## ルーテッド L3 インターフェイス

MTU 値は、L2 ネットワークに接続されているすべてのデバイスで同じにする必要があります。同じでない場合、次の症状が報告される可能性があります。

- Intermediate System-to-Intermediate System (IS-IS) の隣接関係が確立されない。デフォルトでは、IS-ISはhelloパディングを使用します。そのため、あるルータのMTU値が他のルータの値よりも低い場合、helloはジャイアントと見なされ、廃棄される可能性があります。
- Open Shortest Path First (OSPF) の隣接関係が Exstart または Exchange の状態のままになる。これは大きな Database Descriptor (DBD) パケットが giant と見なされ、ドロップされる可能性があるためです。MTU 値が低いルータがパケットを受信すると、データベースは同

期されません。

- 送信デバイスよりも MTU 値が低いデバイスがデータトラフィックを受信すると、データトラフィックが giant と見なされ、ドロップされる。
- 大量の packets がドロップされるときにスループットが低下する。パス MTU ディスカバリでは、大量の packets がドロップされたときに TCP セッションを回復できますが、これはスループットに影響します。

## デフォルト MTU

このセクションでは、mtu コマンドが設定されていない場合の、ルーテッド インターフェイスのデフォルトの MTU を分析します。

```
RP/0/RP0/CPU0:motorhead#sh run int gigabitEthernet 0/1/0/3
interface GigabitEthernet0/1/0/3
cdp
ipv4 address 10.0.1.1 255.255.255.0
ipv6 address 2001:db8::1/64
!
```

```
RP/0/RP0/CPU0:router#sh int gigabitEthernet 0/1/0/3 | i MTU
MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router#show im database interface gigabitEthernet 0/1/0/3
```

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party,  
LDP - Local Data Plane, GDP - Global Data Plane, RED - Redundancy

Node 0/1/CPU0 (0x11)

```
Interface GigabitEthernet0/1/0/3, ifh 0x01180100 (up, 1514)
Interface flags: 0x000000000010059f (IFCONNECTOR|IFINDEX
|SUP_NAMED_SUB|BROADCAST|CONFIG|HW|VIS|DATA
|CONTROL)
Encapsulation: ether
Interface type: IFT_ETHERNET
Control parent: None
Data parent: None
Views: GDP|LDP|L3P|OWN
```

Protocol Caps (state, mtu)

```
-----
None ether (up, 1514)
arp arp (up, 1500)
clns clns (up, 1500)
ipv4 ipv4 (up, 1500)
mpls mpls (up, 1500)
ipv6 ipv6_preswitch (up, 1500)
ipv6 ipv6 (down, 1500)
ether_sock ether_sock (up, 1500)
```

```
RP/0/RP0/CPU0:router#show ipv4 interface gigabitEthernet 0/1/0/3 | i MTU
MTU is 1514 (1500 is available to IP)
```

```
RP/0/RP0/CPU0:router#show ipv6 interface gigabitEthernet 0/1/0/3 | i MTU
MTU is 1514 (1500 is available to IPv6)
```

```
RP/0/RP0/CPU0:router#sh mpls interfaces gigabitEthernet 0/1/0/3 private location 0/1/CPU0
Interface IFH MTU
```

```
-----
Gi0/1/0/3 0x01180100 1500
```

```
RP/0/RP0/CPU0:router#
```

この例では、デフォルトの L2 インターフェイス MTU は 1514 バイトで、これにはイーサネットヘッダーの 14 バイトが含まれています。14 バイトの内訳は、宛先 MAC アドレスが 6 バイト、送信元 MAC アドレスが 6 バイト、タイプまたは長さが 2 バイトです。これには、プリアンブル、フレームデリミタ、フレームチェックシーケンス ( FCS ) の 4 バイト、フレーム間ギャップは含まれません。PPP または HDLC フレームの場合は、L2 ヘッダーの 4 バイトが考慮されるため、デフォルトのインターフェイス MTU は 1504 バイトになります。

L3 プロトコルは、親 MTU のペイロードから MTU を継承します。1514 バイトの L2 MTU から L2 ヘッダーの 14 バイトを引くと、L2 ペイロードは 1500 バイトになります。これが L3 プロトコルの MTU になります。IPv4、IPv6、MPLS、および Connectionless Network Service ( CLNS ) はこの 1500 バイトの MTU を継承します。そのため、Cisco IOS XR イーサネットインターフェイスはデフォルトで 1500 バイトの L3 パケットを転送できます。これは Cisco IOS イーサネットインターフェイスのデフォルトと同じです。

## 非デフォルト MTU

このセクションでは、1500 バイトの IPv4 パケットを、パケットの先頭に 4 バイトの MPLS タグを 2 つ付加して送信するために、1508 バイトの `mpls mtu` を設定する方法を示します。

```
RP/0/RP0/CPU0:router#conf
RP/0/RP0/CPU0:router(config)#int gig 0/1/0/3
RP/0/RP0/CPU0:router(config-if)#mpls mtu 1508
RP/0/RP0/CPU0:router(config-if)#commit
RP/0/RP0/CPU0:Mar 12 00:36:49.807 CET: config[65856]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'root'. Use 'show configuration commit changes 1000000124' to view the
changes.RP/0/RP0/CPU0:router(config-if)#end
RP/0/RP0/CPU0:Mar 12 00:36:54.188 CET: config[65856]: %MGBL-SYS-5-CONFIG_I : Configured
from console by root on vty0 (10.55.144.149)
RP/0/RP0/CPU0:router#sh mpls interfaces gigabitEthernet 0/1/0/3 private location 0/1/CPU0
Interface IFH MTU
-----
Gi0/1/0/3 0x01180100 1500
RP/0/RP0/CPU0:router#show im database interface gigabitEthernet 0/1/0/3

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party,
LDP - Local Data Plane, GDP - Global Data Plane, RED - Redundancy

Node 0/1/CPU0 (0x11)

Interface GigabitEthernet0/1/0/3, ifh 0x01180100 (up, 1514)
Interface flags: 0x0000000000010059f (IFCONNECTOR|IFINDEX
|SUP_NAMED_SUB|BROADCAST|CONFIG|HW|VIS|DATA
|CONTROL)
Encapsulation: ether
Interface type: IFT_ETHERNET
Control parent: None
Data parent: None
Views: GDP|LDP|L3P|OWN

Protocol Caps (state, mtu)
-----
None ether (up, 1514)
arp arp (up, 1500)
clns clns (up, 1500)
ipv4 ipv4 (up, 1500)
mpls mpls (up, 1500)
ipv6 ipv6_preswitch (up, 1500)
```

```
ipv6 ipv6 (down, 1500)
ether_sock ether_sock (up, 1500)
```

```
RP/0/RP0/CPU0:router#
```

**mpls mtu 1508** コマンドはコミットされますが、**show** コマンドの出力で MPLS の MTU が 1500 バイトのままであるため、適用はされていません。これは、L3 子プロトコルは親の L2 インターフェイスのペイロードよりも大きい MTU を持つことができないためです。

1500 バイトの IP パケットの先頭に 2 つのラベルを割り当てるには、次の手順を実行する必要があります。

- すべての子プロトコル ( MPLS を含む ) が 1508 バイトの MTU を継承できるように、1522 バイトの L2 インターフェイス MTU を設定します ( 1522 - 14 = 1508 )。
- MPLS のみが 1500 バイトを超えられるように、L3 プロトコルの MTU を 1500 バイトに減らします。

```
RP/0/RP0/CPU0:router#sh run int gig 0/1/0/3
```

```
interface GigabitEthernet0/1/0/3
cdp
mtu 1522
ipv4 mtu 1500
ipv4 address 10.0.1.1 255.255.255.0
ipv6 mtu 1500
ipv6 address 2001:db8::1/64
!
!
```

```
RP/0/RP0/CPU0:router#show im database interface gigabitEthernet 0/1/0/3
```

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party,  
LDP - Local Data Plane, GDP - Global Data Plane, RED - Redundancy

```
Node 0/1/CPU0 (0x11)
```

```
Interface GigabitEthernet0/1/0/3, ifh 0x01180100 (up, 1522)
Interface flags: 0x000000000010059f (IFCONNECTOR|IFINDEX
|SUP_NAMED_SUB|BROADCAST|CONFIG|HW|VIS|DATA
|CONTROL)
Encapsulation: ether
Interface type: IFT_ETHERNET
Control parent: None
Data parent: None
Views: GDP|LDP|L3P|OWN
```

```
Protocol Caps (state, mtu)
```

```
-----
None ether (up, 1522)
arp arp (up, 1508)
clns clns (up, 1508)
ipv4 ipv4 (up, 1500)
mpls mpls (up, 1508)
ipv6 ipv6_preswitch (up, 1508)
ipv6 ipv6 (down, 1500)
ether_sock ether_sock (up, 1508)
```

```
RP/0/RP0/CPU0:router#
```

この設定により、1500 バイトの IPv4 および IPv6 パケットと 1508 バイトの MPLS パケット (先頭に 2 つのタグが付いた 1500 バイトのパケット) を送信できます。

## ルーテッド L3 サブインターフェイス

ルーテッド L3 サブインターフェイスには次の特性があります。

ルーテッドサブインターフェイスMTUは、親メインインターフェイスのMTUを継承します。サブインターフェイスで設定されたVLANタグごとに4バイトを追加します。したがって、dot1q サブインターフェイスの場合は 4 バイト、IEEE 802.1Q トンネリング ( QinQ ) の場合は 8 バイトあります。

そのため、メイン インターフェイスとサブインターフェイスの両方で同じサイズの L3 パケットを転送できます。

mtu コマンドは、サブインターフェイスで設定できますが、メイン インターフェイスから継承された MTU と等しいか、それよりも小さい場合にのみ適用されます。

これは、メイン インターフェイスの MTU が 2000 バイトの場合の例です。

```
RP/0/RP0/CPU0:router#sh run int gig 0/1/0/3
interface GigabitEthernet0/1/0/3
cdp
mtu 2000
!
```

```
RP/0/RP0/CPU0:router#sh run int gig 0/1/0/3.100
interface GigabitEthernet0/1/0/3.100
ipv4 address 10.0.2.1 255.255.255.0
ipv6 address 2001:db9:0:1::1/64
dot1q vlan 100
!
```

```
RP/0/RP0/CPU0:router#sh int gig 0/1/0/3.100 | i MTU
MTU 2004 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router#show im database interface gigabitEthernet 0/1/0/3.100
```

View: OWN - Owner, L3P - Local 3rd Party, G3P - Global 3rd Party,  
LDP - Local Data Plane, GDP - Global Data Plane, RED - Redundancy

Node 0/1/CPU0 (0x11)

```
Interface GigabitEthernet0/1/0/3.100, ifh 0x01180260 (up, 2004)
Interface flags: 0x0000000000000597 (IFINDEX|SUP_NAMED_SUB
|BROADCAST|CONFIG|VIS|DATA|CONTROL)
Encapsulation: dot1q
Interface type: IFT_VLAN_SUBIF
Control parent: GigabitEthernet0/1/0/3
Data parent: GigabitEthernet0/1/0/3
Views: GDP|LDP|L3P|OWN
```

Protocol Caps (state, mtu)

```
-----
None vlan_jump (up, 2004)
None dot1q (up, 2004)
arp arp (up, 1986)
ipv4 ipv4 (up, 1986)
ipv6 ipv6_preswitch (up, 1986)
ipv6 ipv6 (down, 1986)
```

```
RP/0/RP0/CPU0:router#
```

showコマンドでは、サブインターフェイスのMTUは2004です。サブインターフェイスの下に1つのdot1qタグが設定されているため、メインインターフェイスのMTUに4バイトを追加します。

ただし、IPv4 および IPv6 パケットの MTU は、メイン インターフェイスの MTU と同じままです ( 1986 )。これは、L3プロトコルのMTUが $2004 - 14 - 4 = 1986$ として計算されるようになったためです。

mtu コマンドはサブインターフェイスで設定できますが、設定した MTU はメイン インターフェイスから継承された MTU と等しいか、それよりも小さい場合にのみ適用されます ( メイン インターフェイスの MTU より 4 バイト大きい )。

サブインターフェイスの MTU が、継承された MTU よりも大きい場合は、次に示すようにその MTU は適用されません。

```
RP/0/RP0/CPU0:router#sh int gig 0/1/0/3.100 | i MTU
MTU 2004 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router#conf
RP/0/RP0/CPU0:router(config)#int gig 0/1/0/3.100
RP/0/RP0/CPU0:router(config-subif)#mtu 2100
RP/0/RP0/CPU0:router(config-subif)#commit
RP/0/RP0/CPU0:router(config-subif)#end
RP/0/RP0/CPU0:router#sh int gig 0/1/0/3.100 | i MTU
MTU 2004 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router#
```

そのため、mtu コマンドは、メイン インターフェイスから継承された MTU の値を小さくするためにのみ使用できます。

同様に、サブインターフェイス L2 のペイロードから継承された L3 MTU の値を小さくするために、L3 プロトコル ( IPv4、IPv6、MPLS ) の MTU コマンドを使用することもできます。L3 プロトコルの MTU は、L2 MTU のペイロードに収まらない値に設定されている場合は、有効ではありません。

## L2VPN L2 インターフェイス

擬似回線 ( PW ) の両側の接続回路の MTU が同じでない場合、Label Distribution Protocol ( LDP ) によって PW が起動されないため、L2VPN の MTU が重要になります。

次の show コマンドの出力は、MTU の不一致のために L2VPN PW が起動されない状態を示しています。

```
RP/0/RP0/CPU0:router1#sh l2vpn xconnect
Legend: ST = State, UP = Up, DN = Down, AD = Admin Down, UR = Unresolved,
SB = Standby, SR = Standby Ready, (PP) = Partially Programmed
```

```
XConnect Segment 1 Segment 2
Group Name ST Description ST Description ST
-----
mtu mtu DN Gi0/0/0/2.201 UP 10.0.0.12 201 DN
-----
```

```
RP/0/RP0/CPU0:router1#sh l2vpn xconnect detail
```

```
Group mtu, XC mtu, state is down; Interworking none
AC: GigabitEthernet0/0/0/2.201, state is up
```

```
Type VLAN; Num Ranges: 1
VLAN ranges: [201, 201]
MTU 2000; XC ID 0x1080001; interworking none
Statistics:
packets: received 0, sent 0
bytes: received 0, sent 0
drops: illegal VLAN 0, illegal length 0
PW: neighbor 10.0.0.12, PW ID 201, state is down ( local ready )
PW class mtu-class, XC ID 0xffffe0001
Encapsulation MPLS, protocol LDP
Source address 10.0.0.2
PW type Ethernet, control word disabled, interworking none
PW backup disable delay 0 sec
Sequencing not set
```

```
PW Status TLV in use
MPLS Local Remote
```

```
-----
Label 16046 16046
Group ID 0x1080100 0x6000180
Interface GigabitEthernet0/0/0/2.201 GigabitEthernet0/1/0/3.201
MTU 2000 1986
Control word disabled disabled
PW type Ethernet Ethernet
VCCV CV type 0x2 0x2
(LSP ping verification) (LSP ping verification)
VCCV CC type 0x6 0x6
(router alert label) (router alert label)
(TTL expiry) (TTL expiry)
-----
```

```
Incoming Status (PW Status TLV):
Status code: 0x0 (Up) in Notification message
Outgoing Status (PW Status TLV):
Status code: 0x0 (Up) in Notification message
MIB cpwVcIndex: 4294836225
Create time: 18/04/2013 16:20:35 (00:00:37 ago)
Last time status changed: 18/04/2013 16:20:43 (00:00:29 ago)
```

**Error: MTU mismatched**

```
Statistics:
packets: received 0, sent 0
bytes: received 0, sent 0
RP/0/RP0/CPU0:router1#
RP/0/RP0/CPU0:router1#sh int GigabitEthernet0/0/0/2 | i MTU
MTU 2014 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router1#sh int GigabitEthernet0/0/0/2.201 | i MTU
MTU 2018 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router1#
```

この例では、PW を起動させるために、両側の MPLS L2VPN プロバイダー エッジ ( PE ) が同じ MTU 値を送出する必要がある点に注意してください。

MPLS LDP で送出された MTU には L2 オーバーヘッドが含まれていません。これは、L2 オーバーヘッドを含む XR インターフェイスの **config および show コマンドとは異なります**。サブインターフェイスの MTU は 2018 バイトですが ( 2014 バイトのメイン インターフェイスから継承されるため )、LDP は 2000 バイトの MTU を送出しました。したがって、L2 ヘッダーから 18 バイト ( イーサネット ヘッダーの 14 バイト + 1 つの dot1q タグの 4 バイト ) を引きます。

MTU の不一致を修正するには、各デバイスが接続回線の MTU 値をどのように計算するかを理解することが重要です。これは、ベンダー、プラットフォーム、ソフトウェア バージョン、設定などのパラメータに依存します。

## EVC ( ASR9000 )

Cisco ASR 9000 シリーズ アグリゲーション サービス ルータは、EVC インフラストラクチャ モードを使用します。これにより、L2VPN L2 インターフェイスおよびサブインターフェイスで柔軟に VLAN を一致させることができます。

EVC L2VPN L2 インターフェイスには次の特性があります。

- **encapsulation** コマンドで 1 つ以上のタグを設定できます。
- デフォルトでは、**encapsulation** コマンドだけでタグが保持され、PW を介して伝送されます。このため、デフォルトではタグを削除する必要はありません ( 非 EVC プラットフォームでは削除する必要があります )。
- 着信タグをポップするときや、着信フレームの先頭にいくつかの追加タグをプッシュするときは、**rewrite** コマンドを使用します。

サブインターフェイス MTU を計算するには、メイン インターフェイス MTU ( デフォルトまたはメイン インターフェイスで手動で設定されたもの ) を取得し、**encapsulation** コマンドで設定された VLAN タグごとに 4 バイトを追加します。「[特定の EFP Encapsulation コマンド](#)」を参照してください。

サブインターフェイスで **mtu** コマンドが設定されている場合は、計算された MTU より小さい場合にのみ有効です。**rewrite** コマンドは、サブインターフェイス MTU に影響しません。

以下が一例です。

```
RP/0/RSP0/CPU0:router2#sh run int gig 0/1/0/3
interface GigabitEthernet0/1/0/3
cdp
mtu 2014
negotiation auto
!
```

```
RP/0/RSP0/CPU0:router2#sh run int gig 0/1/0/3.201
interface GigabitEthernet0/1/0/3.201 l2transport
encapsulation dot1q 201 second-dot1q 10
rewrite ingress tag pop 2 symmetric
!
```

```
RP/0/RSP0/CPU0:router2#
RP/0/RSP0/CPU0:router2#sh int gig 0/1/0/3.201
GigabitEthernet0/1/0/3.201 is up, line protocol is up
Interface state transitions: 1
Hardware is VLAN sub-interface(s), address is 0024.986c.63f3
Layer 2 Transport Mode
MTU 2022 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
```

この例では、メインインターフェイスのMTUは2014バイトです。サブインターフェイスの下に2つのタグが設定されているため、8バイトを追加します。

サブインターフェイスで**mtu 2026**バイトを設定した場合、メインインターフェイス(2022)から継承されたサブインターフェイスのMTUよりも大きいため、このバイトは適用されません。この結果、設定できるのは2022バイトよりも小さいサブインターフェイスMTUだけです。

このサブインターフェイス MTU に基づいて、ネイバーに送出される MPLS LDP ペイロードの MTU を計算し、リモート L2VPN PE によって計算されたものと同じであることを確認します。ここで **rewrite** コマンドがその役割を果たします。

MPLS LDP ペイロードの MTU を計算するには、サブインターフェイスの MTU を取得して、次の手順を実行します。

1. イーサネット ヘッダーの 14 バイトを引きます。
2. サブインターフェイスで設定された `rewrite` コマンドでポップされるタグごとに 4 バイトを引きます。
3. サブインターフェイスで設定された `rewrite` コマンドでプッシュされるタグごとに 4 バイトを加えます。

これは、`gig 0/1/0/3.201` での QinQ 設定と同じ例です。

```
interface GigabitEthernet0/1/0/3
cdp
mtu 2014
negotiation auto
!
interface GigabitEthernet0/1/0/3.201 l2transport
encapsulation dot1q 201 second-dot1q 10
rewrite ingress tag pop 2 symmetric
!

RP/0/RSP0/CPU0:router2#sh int gig 0/1/0/3.201
GigabitEthernet0/1/0/3.201 is up, line protocol is up
Interface state transitions: 1
Hardware is VLAN sub-interface(s), address is 0024.986c.63f3
Layer 2 Transport Mode
MTU 2022 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
```

次に、MPLS LDP ペイロードの MTU を計算します。

1. サブインターフェイス MTU の MTU 値 : 2022 バイト
2. イーサネット ヘッダーの 14 バイトを引く :  $2022 - 14 = 2008$  バイト
3. 各ポップタグの 4 バイトを引きます。書き換え コマンド :  $2008 - 4 * 2 = 2000$

リモート側が 2000 バイトの MPLS LDP ペイロードをアドバタイズすることを確認します。そうでない場合は、ローカルまたはリモートの接続回線 (AC) の MTU サイズを調整して一致させます。

```
RP/0/RSP0/CPU0:router2#sh l2vpn xconnect det

Group mtu, XC mtu, state is up; Interworking none
AC: GigabitEthernet0/1/0/3.201, state is up
Type VLAN; Num Ranges: 1
Outer Tag: 201
VLAN ranges: [10, 10]
MTU 2000; XC ID 0x1880003; interworking none
```

特定のイーサネット フロー ポイント ( EFP ) Encapsulation コマンド

これらのカプセル化は、どのタグにも一致しないものと見なされるため、サブインターフェイス MTU を増やしません。

- `encapsulation untagged`
- `encapsulation default`

これらのカプセル化の修飾子は、サブインターフェイス MTU を計算するために必要なタグの数に影響しません。

- ネイティブ
- payload-ethertype
- exact
- cos
- ingress source-mac または ingress destination-mac

**encapsulation [dot1q|dot1ad] priority-tagged は、1 つのタグに一致するものと見なされます。**

最も深いタグ一致として使用される「any」キーワードは、サブインターフェイス MTU を増やしません。

- **encapsulation dot1q any は、サブインターフェイス MTU を増やしません。**
- **encapsulation dot1ad 10 dot1q any は1つのタグと見なされ、サブインターフェイスMTUが4バイト増加します。**
- **encapsulation dot1ad any dot1q 7は2つのタグと見なされ、サブインターフェイスMTUが8バイト増加します。**

VLAN ID の範囲は、サブインターフェイス MTU を増やします。

- **encapsulation dot1q 10-100は1つのタグと見なされ、サブインターフェイスMTUを4バイト増やします。**

分離性一致である EFP のカプセル化 MTU オーバーヘッドは、最大要素の MTU として扱われます。

- **encapsulation dot1q 10-100, untagged は、範囲 10 ~ 100 が最大要素であるため、1 つのタグと見なされます。**

## 非 EVC ( XR 12000 および CRS )

Cisco XR 12000 シリーズ ルータや Carrier Routing System ( CRS ) などのルータでは、サブインターフェイスでの VLAN 照合に従来の設定を使用します。EVC モデルに従わない CRS および XR 12000 ルータの L2VPN L2 インターフェイスには、次の特性があります。

- 非 EVC プラットフォームでは、着信 dot1q または dot1ad タグは、L2 トランスポート サブインターフェイスで受信されると自動的に削除されます。
- 送出する MPLS LDP のペイロード サイズを計算するときには、**show interface** コマンドで表示されるように、サブインターフェイスの MTU からタグのサイズを引きます。
- これはルーテッド サブインターフェイスに似ています。
- サブインターフェイスはメインインターフェイスから MTU を継承します。サブインターフェイスの MTU を計算するには、メインインターフェイスの MTU にタグごとに 4 バイトを追加します。たとえば、QinQ サブインターフェイスに 2 つの dot1q タグがある場合、サブインターフェイスの MTU はデフォルトでメイン インターフェイスの MTU よりも 8 バイト大きくなります。
- また、サブインターフェイスで **mtu** コマンドを使用できます。ただし、このコマンドは、メイン インターフェイスの MTU から継承されたサブインターフェイスの MTU を減らすためにのみ使用します。

これらの特性を説明する例を次にいくつか示します。

次の例では、非 EVC サブインターフェイスの設定方法を示します。

```
RP/0/RP0/CPU0:router1#sh run int gigabitEthernet 0/0/0/2.201
interface GigabitEthernet0/0/0/2.201 l2transport
dot1q vlan 201
!
```

```
RP/0/RP0/CPU0:router1#
```

非 EVC プラットフォームは、EVC プラットフォーム ( ASR9000 ) の encapsulation および rewrite コマンドの代わりに、dot1q vlan または dot1ad vlan コマンドを使用します。

メイン インターフェイスまたはサブインターフェイスで MTU を明示的に設定しない場合は、デフォルトで 1500 バイトの L3 パケットを受信できます。

```
RP/0/RP0/CPU0:router1#sh int gig 0/0/0/2 | i MTU
MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router1#sh int gig 0/0/0/2.201 | i MTU
MTU 1518 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router1#
```

サブインターフェイス MTU はメインインターフェイス MTU (1514) から計算されます。各 dot1q タグに 4 バイトを追加します。dot1q vlan 201 コマンドでサブインターフェイスにタグが 1 つ設定されているため、1514 バイトに 4 バイトを追加し、1518 バイトの MTU にします。

MPLS LDP の対応するペイロード MTU は 1500 バイトです。イーサネット ヘッダーの 14 バイトがカウントされず、1 つの dot1q タグが PW を通過するとき非 EVC プラットフォームによって自動的にポップされるためです。

```
RP/0/RP0/CPU0:router1#sh l2vpn xconnect detail
```

```
Group mtu, XC mtu, state is down; Interworking none
AC: GigabitEthernet0/0/0/2.201, state is up
Type VLAN; Num Ranges: 1
VLAN ranges: [201, 201]
MTU 1500; XC ID 0x1080001; interworking none
```

メイン インターフェイスの MTU を 2014 バイトに増やすと、サブインターフェイスの MTU はそれに応じて増加します。

```
RP/0/RP0/CPU0:router1#sh run int gig 0/0/0/2
interface GigabitEthernet0/0/0/2
description static lab connection to head 4/0/0 - dont change
cdp
mtu 2014
ipv4 address 10.0.100.1 255.255.255.252
load-interval 30
!
```

```
RP/0/RP0/CPU0:router1#sh run int gig 0/0/0/2.201
interface GigabitEthernet0/0/0/2.201 l2transport
dot1q vlan 201
!
```

```
RP/0/RP0/CPU0:router1#sh int gig 0/0/0/2 | i MTU
MTU 2014 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router1#sh int gig 0/0/0/2.201 | i MTU
MTU 2018 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
RP/0/RP0/CPU0:router1#sh l2vpn xconnect detail
```

```
Group mtu, XC mtu, state is down; Interworking none
```

```
AC: GigabitEthernet0/0/0/2.201, state is up
Type VLAN; Num Ranges: 1
VLAN ranges: [201, 201]
MTU 2000; XC ID 0x1080001; interworking none
```

したがって、MPLS LDP の MTU を計算するには、イーサネット ヘッダーの 14 バイトを引き、サブインターフェイスで設定されているタグごとに 4 バイトを追加します。

## 自動イーサネット インターフェイス ドライバ MTU および MRU の設定

イーサネット インターフェイスでは、インターフェイス ドライバに、インターフェイス MTU の設定に基づく MTU および MRU が設定されます。

イーサネット インターフェイス ドライバに設定されている MTU および MRU は、`show controller <interface> all` コマンドを使用して表示できます。

Cisco IOS XR リリース 5.1.1 よりも前のリリースでは、イーサネット インターフェイス ドライバの MTU および MRU は、インターフェイスの Cisco IOS XR MTU 設定に基づいて自動的に設定されました。

イーサネット ドライバの MTU/MRU は、単に設定済みの MTU と、2 つの追加のイーサネット タグと CRC フィールドに対応する 12 バイトに基づいて設定されました。この 12 バイトは、サブインターフェイスで VLAN タグが設定されているかどうかにかかわらず、イーサネット ドライバ MTU/MRU に追加されました。

Cisco IOS XR リリース 5.1.1 よりも前のすべての Cisco IOS XR バージョンと ASR 9000 インターフェイスでの 1514 バイトのデフォルト MTU の例を次に示します。

```
RP/0/RSP0/CPU0:ASR2#show interface Gi0/2/0/0
GigabitEthernet0/2/0/0 is up, line protocol is up
  Interface state transitions: 3
  Hardware is GigabitEthernet, address is 18ef.63e2.0598 (bia 18ef.63e2.0598)
  Description: Static_Connections_to_ME3400-1_Gi_0_2 - Do Not Change
  Internet address is Unknown
  MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
<snip>
```

MTU/MRU programmed on ethernet interface driver is 1514 + 12 bytes

```
RP/0/RSP0/CPU0:ASR2#show controllers Gi0/2/0/0 all
```

```
<snip>
Operational values:
  Speed: 1Gbps
  Duplex: Full Duplex
  Flowcontrol: None
  Loopback: None (or external)
  MTU: 1526
  MRU: 1526
  Inter-packet gap: standard (12)
<snip>
```

Cisco IOS XR リリース 5.1.1 以降では、イーサネット インターフェイス ドライバで使用される MTU および MRU が変更されており、いずれかのサブインターフェイスで設定されている VLAN

タグの数に基づいて設定されます。

どのサブインターフェイスでも VLAN タグが設定されていない場合、ドライバの MTU/MRU はインターフェイスの設定済み MTU + 4 CRC バイトに等しくなります (たとえば、 $1514 + 4 = 1518$  バイト)。

いずれかのサブインターフェイスで VLAN タグが 1 つ設定されている場合、ドライバの MTU/MRU は設定済み MTU + 8 バイト (1 タグ + CRC) に等しくなります (たとえば、 $1514 + 8 = 1522$  バイト)。

いずれかのサブインターフェイスで VLAN タグが 2 つ設定されている場合、ドライバの MTU/MRU は設定済み MTU + 12 バイト (1 タグ + CRC) に等しくなります (たとえば、 $1514 + 12 = 1526$  バイト)。

**any** キーワードを含む QinQ で **second-dot1q** タグが設定されている場合、ドライバの MTU/MRU は設定済み MTU + 8 バイト (1 タグ + CRC) に等しくなります (たとえば、 $1514 + 8 = 1522$  バイト)。

次の例は、ASR 9000 での Cisco IOS XR リリース 5.1.1 以降の動作を示しています。

```
RP/0/RSP0/CPU0:ASR2#sh run int ten0/1/0/0
interface TenGigE0/1/0/0
  cdp
```

```
RP/0/RSP0/CPU0:ASR2#show controllers ten0/1/0/0 all
```

```
<snip>
Operational values:
  Speed: 10Gbps
  Duplex: Full Duplex
  Flowcontrol: None
  Loopback: Internal
  MTU: 1518
  MRU: 1518
  Inter-packet gap: standard (12)
<snip>
```

```
RP/0/RSP0/CPU0:ASR2#config
RP/0/RSP0/CPU0:ASR2(config-if)#int ten0/1/0/0.1
RP/0/RSP0/CPU0:ASR2(config-subif)#encapsulation dot1q 1
RP/0/RSP0/CPU0:ASR2(config-subif)#commit
```

```
RP/0/RSP0/CPU0:ASR2#show controllers ten0/1/0/0 all
```

```
<snip>
Operational values:
  Speed: 10Gbps
  Duplex: Full Duplex
  Flowcontrol: None
  Loopback: Internal
  MTU: 1522
  MRU: 1522
  Inter-packet gap: standard (12)
<snip>
```

```
RP/0/RSP0/CPU0:ASR2#config
RP/0/RSP0/CPU0:ASR2(config)#int ten0/1/0/0.2
RP/0/RSP0/CPU0:ASR2(config-subif)#encapsulation dot1q 10 second-dot1q 20
```

```
RP/0/RSP0/CPU0:ASR2(config-subif)#commit
```

```
RP/0/RSP0/CPU0:ASR2#show controllers ten0/1/0/0 all
```

```
<snip>
```

```
Operational values:
```

```
Speed: 10Gbps
```

```
Duplex: Full Duplex
```

```
Flowcontrol: None
```

```
Loopback: Internal
```

```
MTU: 1526
```

```
MRU: 1526
```

```
Inter-packet gap: standard (12)
```

```
<snip>
```

```
RP/0/RSP0/CPU0:ASR2#config
```

```
RP/0/RSP0/CPU0:ASR2(config)#int ten0/2/0/0
```

```
RP/0/RSP0/CPU0:ASR2(config)#cdp
```

```
RP/0/RSP0/CPU0:ASR2(config)#int ten0/2/0/0.1 l2transport
```

```
RP/0/RSP0/CPU0:ASR2(config-subif)#encapsulation dot1q 10 second-dot1q any
```

```
RP/0/RSP0/CPU0:ASR2(config-subif)#commit
```

```
RP/0/RSP0/CPU0:ASR2#show controllers ten0/1/0/0 all
```

```
<snip>
```

```
Operational values:
```

```
Speed: 10Gbps
```

```
Duplex: Full Duplex
```

```
Flowcontrol: None
```

```
Loopback: Internal
```

```
MTU: 1522
```

```
MRU: 1522
```

```
Inter-packet gap: standard (12)
```

```
<snip>
```

リリース 5.1.1 以降でのこの動作変更は、多くの場合、インターフェイスでの MTU の設定変更を必要としません。

この動作変更は、単一の VLAN タグが設定されているサブインターフェイスで 2 つの VLAN タグを持つパケットを受信した場合に問題を引き起こす可能性があります。この状況では、受信したパケットがイーサネット インターフェイス ドライバの MRU を超える可能性があります。この問題を回避するには、インターフェイス MTU を 4 バイト増やすか、サブインターフェイスに VLAN タグを 2 つ設定します。

リリース 5.1.1 におけるイーサネット インターフェイス ドライバの MTU および MRU の自動設定は、CRS ルータと ASR 9000 ルータで同じです。ただし、リリース 5.1.1 を実行する CRS ルータでは、**show controller** 出力で表示される MTU および MRU 値に 4 バイトの CRC が含まれません。MTU および MRU の表示方法は、CRS と ASR9000 で異なります。

```
RP/0/RP0/CPU0:CRS#sh run int ten0/4/0/0
```

```
Mon May 19 08:49:26.109 UTC
```

```
interface TenGigE0/4/0/0
```

```
<snip>
```

```
Operational values:
```

```
Speed: 10Gbps
```

```
Duplex: Full Duplex
```

```
Flowcontrol: None
```

```
Loopback: None (or external)
```

```
MTU: 1514
```

```
MRU: 1514
```

```
Inter-packet gap: standard (12)
```

```
RP/0/RP0/CPU0:CRS(config)#int ten0/4/0/0.1
RP/0/RP0/CPU0:CRS(config-subif)#encapsulation dot1q 1
RP/0/RP0/CPU0:CRS(config-subif)#commit
```

Operational values:

```
Speed: 10Gbps
Duplex: Full Duplex
Flowcontrol: None
Loopback: None (or external)
MTU: 1518
MRU: 1518
Inter-packet gap: standard (12)
```

ASR 9000 で MTU および MRU が show controller 出力に表示される方法は、表示される MTU/MRU 値に 4 バイトの CRC が含まれないように将来的に変更される予定です。この将来的な変更は、Cisco bug ID [CSCuo93379](#) で追跡できます。

## リリース 5.1.1 よりも前のリリースからリリース 5.1.1 以降にアップグレードする際の設定の変換

- デフォルト MTU :

リリース 5.1.1 より前のリリースで、メイン インターフェイスがあり、サブインターフェイスと mtu コマンドがない場合 :

```
interface TenGigE0/1/0/19
l2transport
!
```

また、このインターフェイスで dot1q または QinQ フレームを伝送する場合、リリース 5.1.1 以降では、MTU を「mtu 1522」に手動で設定する必要があります。

```
interface TenGigE0/1/0/19
mtu 1522
l2transport
!
```

この設定により、QinQ フレームを以前のリリースと同様に伝送できるようになります。伝送するのが dot1q のみで QinQ は伝送しない場合は、MTU 値を 1518 に設定できます。

リリース 5.1.1 より前のリリースで、dot1q または QinQ が設定され、「any」キーワードを使用するサブインターフェイスがあり、2 つの明示的なタグが付いた QinQ サブインターフェイスが設定されていない場合。

```
interface TenGigE0/1/0/19
!
interface TenGigE0/1/0/19.100 l2transport
encapsulation dot1q 100
!
interface TenGigE0/1/0/19.101 l2transport
encapsulation dot1q 101 second-dot1q any
!
```

リリース 5.1.1 以降のこの設定では、1 つのタグが付いたフレームの伝送のみが許可されるので、QinQ フレームを伝送する場合は、MTU を手動で 4 バイト増やす必要があります。

```
interface TenGigE0/1/0/19
mtu 1518
!
interface TenGigE0/1/0/19.100 l2transport
encapsulation dot1q 100
!
interface TenGigE0/1/0/19.101 l2transport
encapsulation dot1q 101 second-dot1q any
!
```

2 つの明示的なタグが付いた QinQ サブインターフェイス ( 「any」 キーワードを使用しない ) が設定されている場合、リリース 5.1.1 以降にアップグレードするときに、MTU 設定を変更する必要はありません。

```
interface TenGigE0/1/0/19
!
interface TenGigE0/1/0/19.101 l2transport
encapsulation dot1q 101 second-dot1q 200
!
```

L2 トランスポート サブインターフェイスがなく、L3 ルーテッド インターフェイスのみがある場合、MTU 設定が両側で一致し、MTU よりも大きいフレームは伝送されないと考えられます。リリース 5.1.1 以降にアップグレードするときに、MTU 設定を更新する必要はありません。

- リリース 5.1.1 より前のリリースにおけるデフォルト以外の MTU :

同様に、リリース 5.1.1 より前のリリースでデフォルト以外の MTU が設定されていて、サブインターフェイスが設定されておらず、dot1q または QinQ フレームを伝送する必要がある場合は、リリース 5.1.1 以降にアップグレードするときに、設定されている MTU 値を 8 バイト増やす必要があります。

リリース 5.1.1 より前のリリース :

```
interface TenGigE0/1/0/19
mtu 2000
l2transport
!
```

リリース 5.1.1 以降にアップグレードするときに、MTU を手動で 8 バイト増やす必要があります。

```
interface TenGigE0/1/0/19
mtu 2008
l2transport
!
```

dot1q サブインターフェイスがあつて QinQ サブインターフェイスがない場合、または second-dot1q タグに any キーワードを使用する QinQ サブインターフェイスがある場合は、設定されている MTU 値を 4 バイト増やす必要があります。

リリース 5.1.1 より前のリリース :

```
interface TenGigE0/1/0/19
mtu 2000
!
interface TenGigE0/1/0/19.100 l2transport
encapsulation dot1q 100
!
interface TenGigE0/1/0/19.101 l2transport
encapsulation dot1q 101 second-dot1q any
!
```

リリース 5.1.1 以降：

```
interface TenGigE0/1/0/19
mtu 2004
!
interface TenGigE0/1/0/19.100 l2transport
encapsulation dot1q 100
!
interface TenGigE0/1/0/19.101 l2transport
encapsulation dot1q 101 second-dot1q any
!
```

2つの明示的なタグが付いた QinQ サブインターフェイス（「any」キーワードを使用しない）が設定されている場合、リリース 5.1.1 以降にアップグレードするときに、MTU 設定を変更する必要はありません。

```
interface TenGigE0/1/0/19
!
interface TenGigE0/1/0/19.101 l2transport
encapsulation dot1q 101 second-dot1q 200
!
```

L2 トランスポート サブインターフェイスがなく、L3 ルーテッド インターフェイスのみがある場合、MTU 設定が両側で一致し、MTU よりも大きいフレームは伝送されないと考えられます。リリース 5.1.1 以降にアップグレードするときに、MTU 設定を更新する必要はありません。

## 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。