

Cisco IOS XR 및 Cisco IOS 라우터의 MTU 동작

목차

[소개](#)

[배경 정보](#)

[구성](#)

[Cisco IOS 및 Cisco IOS XR 소프트웨어 비교](#)

[라우티드 L3 인터페이스](#)

[기본 MTU](#)

[기본이 아닌 MTU](#)

[라우티드 L3 하위 인터페이스](#)

[L2VPN L2 인터페이스](#)

[EVC\(ASR9000\)](#)

[비 EVC\(XR 12000 및 CRS\)](#)

[자동 이더넷 인터페이스 드라이버 MTU 및 MRU 구성](#)

[릴리스 5.1.1 이전 릴리스에서 릴리스 5.1.1 이상으로 업그레이드할 때 컨피그레이션 변환](#)

소개

이 문서에서는 Cisco IOS® XR 라우터의 MTU(Maximum Transmission Unit) 동작을 설명하고 이러한 동작을 Cisco IOS 라우터와 비교합니다. 또한 EVC(Ethernet Virtual Connection) 및 비 EVC 모델을 모두 사용하는 라우티드 L3(Layer 3) 인터페이스 및 L2VPN(Layer 2 VPN) L2 인터페이스의 MTU에 대해서도 설명합니다. 또한 이 문서에서는 릴리스 5.1.1 이상에서 이더넷 인터페이스 드라이버 MTU 및 MRU(Maximum Receive Unit)가 자동으로 구성되는 방식에 대한 중요한 변경 사항을 설명합니다.

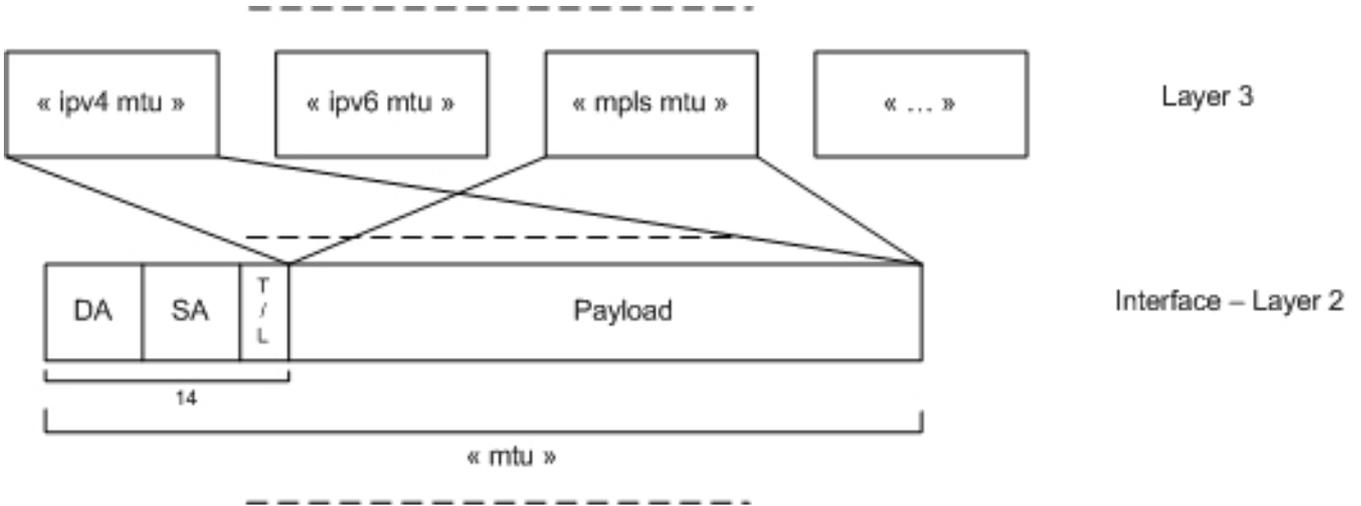
배경 정보

컴퓨터 네트워킹에서 레이어의 통신 프로토콜의 MTU는 레이어가 하나의 인터페이스를 통해 전송할 수 있는 최대 프로토콜 데이터 유닛의 크기(바이트)를 정의합니다. 하나의 MTU 매개변수가 각 인터페이스, 레이어 및 프로토콜과 연결됩니다.

Cisco IOS XR 소프트웨어의 MTU 특성은 다음과 같습니다.

- MTU config 및 show 명령(L2 및 L3)에는 해당 레이어의 헤더 크기가 포함됩니다. 예를 들어, L2 MTU를 구성하는 **mtu** 명령은 이더넷 인터페이스(dot1q 제외)에 14바이트, PPP(Point-to-Point Protocol) 또는 HDLC(high-level data link control)에 4바이트를 포함합니다. **ipv4 mtu** 명령은 IPv4 헤더의 20바이트를 포함합니다.
- 상위 레이어의 MTU는 하위 레이어의 페이로드 내에 들어가야 합니다. 예를 들어, non-dot1q 이더넷 인터페이스의 인터페이스 MTU가 기본값인 1514바이트라면 MPLS(Multiprotocol Label

Switching)와 같은 상위 레이어 프로토콜은 해당 인터페이스에서 최대 MTU가 1500바이트가 될 수 있습니다. 즉, 이더넷 프레임 내에 1500바이트 MPLS 프레임(레이블 포함)만 넣을 수 있습니다. 1500바이트 IPv4 패킷 위에 두 개의 MPLS 태그를 허용하려는 경우 해당 인터페이스에서 1508바이트 MPLS MTU를 구성할 수 없습니다. 이더넷 인터페이스에서 1508바이트 MPLS 프레임을 전송하려면 인터페이스 MTU를 1522 이상의 값으로 늘려야 L2 인터페이스 페이로드가 MPLS 프레임을 전달할 수 있을 만큼 커집니다.



- Cisco IOS XR 소프트웨어가 아닌 기존 Cisco IOS 소프트웨어에서 interface mtu 명령은 L2 페이로드 크기를 구성하지만 L2 헤더는 포함하지 않습니다. 이는 interface mtu 명령에 L2 및 L3 오버헤드가 모두 포함된 Cisco IOS XR 소프트웨어와 다릅니다. L3 MTU 명령은 ipv4 mtu 명령의 경우와 마찬가지로 L3 헤더를 포함하는 프로토콜의 최대 패킷 크기를 구성합니다. 이는 Cisco IOS XR 소프트웨어의 경우와 유사합니다.
- Cisco IOS XR 소프트웨어의 기본 인터페이스 MTU는 1500바이트 L3 패킷의 전송을 허용해야 합니다. 따라서 기본 MTU는 기본 이더넷 인터페이스의 경우 1514바이트이고 직렬 인터페이스의 경우 1504바이트입니다.

이 문서의 나머지 부분에서는 MTU 특성을 설명하고 Cisco IOS와 Cisco IOS XR 소프트웨어 동작을 비교하며 이러한 인터페이스 유형의 예를 제공합니다.

- 라우터 L3 인터페이스
- 라우터 L3 하위 인터페이스
- L2VPN L2 인터페이스

구성

참고: 이 섹션에서 사용된 [명령어](#)에 대한 자세한 내용을 보려면 [Command Lookup Tool](#)([등록된 고객만 해당](#))을 사용하십시오.

참고: Output [Interpreter Tool](#)([등록된 고객만 해당](#))은 특정 show 명령을 지원합니다. show 명령 출력의 분석을 보려면 아웃풋 인터프리터 툴을 사용합니다.

Cisco IOS 및 Cisco IOS XR 소프트웨어 비교

이 섹션에서는 Cisco IOS 및 Cisco IOS XR 소프트웨어 동작을 MTU 특성을 참조하여 비교합니다.

Cisco IOS 소프트웨어에서 mtu 명령 및 해당 show 명령은 L2 헤더를 포함하지 않습니다. L3 헤더를 포함하여 L3 패킷의 최대 크기로 L2 페이로드를 구성하려면 mtu 명령을 사용합니다.

이는 Cisco IOS XR 소프트웨어와는 다릅니다. mtu 명령은 L2 헤더(이더넷의 경우 14바이트, PPP/HDLC의 경우 4바이트)를 포함합니다.

Cisco IOS 라우터가 mtu x로 구성되어 있고 Cisco IOS XR 라우터에 연결되어 있는 경우, Cisco IOS XR 라우터의 해당 인터페이스는 이더넷 인터페이스에 대해 mtu x+14, 또는 직렬 인터페이스에 대해 mtu x+4로 구성해야 합니다.

Cisco IOS 및 Cisco IOS XR 소프트웨어는 ipv4 mtu, ipv6 mtu 및 mpls mtu 명령에 동일한 의미를 갖습니다. 동일한 값으로 구성해야 합니다.

따라서 이더넷 인터페이스의 Cisco IOS Software에서 다음과 같은 컨피그레이션을 수행합니다.

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

Cisco IOS XR 소프트웨어 인접 디바이스의 해당 컨피그레이션은 다음과 같습니다.

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

라우티드 L3 인터페이스

MTU 값은 L2 네트워크에 연결된 모든 디바이스에서 동일해야 합니다. 그렇지 않으면 다음과 같은 증상이 나타날 수 있습니다.

- IS-IS(Intermediate System-to-Intermediate System) 인접성이 나타나지 않습니다. 기본적으로 IS-IS는 hello-padding을 사용합니다. 따라서 Hello는 자이언트로 특징지어질 수 있으며 한 라우터의 MTU 값이 다른 라우터의 값보다 낮을 경우 삭제될 수 있습니다.
- OSPF(Open Shortest Path First) 인접성이 Exstart 또는 Exchange 상태에서 중단됩니다. DBD(Large Database Descriptor) 패킷이 거인으로 간주되어 삭제될 수 있기 때문입니다. MTU 값이 더 낮은 라우터에서 패킷이 수신되면 데이터베이스가 동기화되지 않습니다.
- 데이터 트래픽은 큰 트래픽 특성을 가지며 MTU 값이 전송 디바이스보다 낮은 디바이스에서 수신될 때 삭제됩니다.
- 대용량 패킷이 삭제될 경우 처리량이 적습니다. 경로 MTU 검색의 경우 대용량 패킷이 삭제될 때 TCP 세션을 복구할 수 있지만, 이는 처리량에 영향을 줍니다.

기본 MTU

이 섹션에서는 mtu 명령이 구성되지 않은 경우 라우티드 인터페이스의 기본 MTU를 분석합니다.

```
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#
```

이 예에서 기본 L2 인터페이스 MTU는 1514바이트이며 이더넷 헤더 14바이트를 포함합니다. 14바이트는 대상 MAC 주소 6바이트, 소스 MAC 주소 6바이트, 유형 또는 길이 2바이트로 설명됩니다. 여기에는 프리앰블, 프레임 구분 기호, 4바이트의 FCS(Frame Check Sequence) 및 프레임 간 간격이 포함되지 않습니다. PPP 또는 HDLC 프레임의 경우 L2 헤더의 4바이트가 고려되므로 기본 인터페이스 MTU는 1504바이트입니다.

L3 하위 프로토콜은 상위 MTU의 페이로드에서 MTU를 상속합니다. 1514바이트의 L2 MTU에서 L2 헤더의 14바이트를 빼면 L2 페이로드는 1500바이트입니다. 이는 L3 프로토콜에 대한 MTU가 됩니다. IPv4, IPv6, MPLS 및 CLNS(Connectionless Network Service)는 이 1500바이트 MTU를 상속합니다. 따라서 Cisco IOS XR 이더넷 인터페이스는 기본적으로 Cisco IOS 이더넷 인터페이스의 기본 값과 동일한 1500바이트 L3 패킷을 전송할 수 있습니다.

기본이 아닌 MTU

이 섹션에서는 1500바이트의 IPv4 패킷을 패킷 위에 각각 4바이트의 2개의 MPLS 태그와 함께 전송하기 위해 MPLS mtu를 1508로 구성하는 방법을 보여줍니다.

```

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: 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#

mpls mtu 1508 명령은 커밋되지만 적용되지 않습니다. MPLS의 MTU는 show 명령에서 1500바이트이기 **때문입니다**. 이는 L3 하위 프로토콜이 상위 L2 인터페이스의 페이로드보다 큰 MTU를 가질 수 없기 **때문입니다**.

1500바이트 IP 패킷 위에 두 개의 레이블을 허용하려면 다음을 수행해야 합니다.

- 모든 하위 프로토콜(MPLS 포함)이 1508바이트(1522 - 14 = 1508)의 MTU를 상속하도록 L2 인터페이스 MTU를 1522바이트로 구성합니다.
- MPLS만 1500바이트를 초과할 수 있도록 L3 프로토콜의 MTU를 1500바이트로 줄입니다.

```

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#
```

이 컨피그레이션을 사용하면 1500바이트의 IPv4 및 IPv6 패킷과 1508바이트의 MPLS 패킷을 전송할 수 있습니다(맨 위에 두 개의 태그가 있는 1500바이트 패킷).

라우터드 L3 하위 인터페이스

이러한 특성은 라우터드 L3 하위 인터페이스에 적용됩니다.

라우터드 하위 인터페이스 MTU는 상위 기본 인터페이스의 MTU를 상속합니다. 하위 인터페이스에 구성된 각 VLAN 태그에 4바이트를 추가합니다. 따라서 dot1q 하위 인터페이스에는 4바이트, IEEE 802.1Q 터널링(QinQ) 하위 인터페이스에는 8바이트가 있습니다.

그 결과, 동일한 크기의 L3 패킷들이 메인 인터페이스 및 서브 인터페이스 양쪽에서 포워딩될 수 있다.

mtu 명령은 하위 인터페이스에서 구성할 수 있지만, 기본 인터페이스에서 상속된 MTU보다 낮거나 같은 경우에만 적용됩니다.

다음은 기본 인터페이스의 MTU가 2000바이트인 예입니다.

```
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: 0x00000000000000597 (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 명령에서 하위 인터페이스의 MTU는 2004입니다. 하위 인터페이스 아래에 dot1q 태그가 하나 구성되어 있으므로 주 인터페이스의 MTU에 4바이트를 추가합니다.

그러나 IPv4 및 IPv6 패킷의 MTU는 여전히 주 인터페이스(1986)의 MTU와 동일합니다. L3 프로토콜의 MTU가 $2004 - 14 - 4 = 1986$ 으로 계산되기 때문입니다.

mtu 명령은 하위 인터페이스에서 구성할 수 있지만, 구성된 MTU는 주 인터페이스에서 상속된 MTU보다 낮거나 같은 경우에만 적용됩니다(주 인터페이스의 MTU보다 4바이트 더 큼).

하위 인터페이스의 MTU가 상속된 MTU보다 클 경우, 여기에 표시된 것처럼 적용되지 않습니다.

```
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#
```

따라서 주 인터페이스에서 상속된 MTU 값을 낮추려면 **mtu** 명령만 사용할 수 있습니다.

마찬가지로, 하위 인터페이스 L2 페이로드에서 상속된 L3 MTU의 값을 낮추기 위해 L3 프로토콜 (IPv4, IPv6, MPLS)의 MTU 명령을 사용할 수도 있습니다. L3 프로토콜 MTU는 L2 MTU의 페이로드에 맞지 않는 값으로 구성된 경우 적용되지 않습니다.

L2VPN L2 인터페이스

L2VPN의 MTU는 PW의 양쪽에 있는 연결 회로의 MTU가 동일하지 않을 때 LDP(Label Distribution Protocol)가 PW(의사 와이어)를 발생시키지 않기 때문에 중요합니다.

다음은 MTU 불일치가 있을 때 L2VPN PW가 중단 상태를 유지하는 것을 보여주는 show 명령입니다.

```
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#

이 예에서는 각 측의 MPLS L2VPN PE(Provider Edge)가 PW를 올리기 위해 동일한 MTU 값에 신호를 보내야 합니다.

MPLS LDP에서 시그널링하는 MTU에는 L2 오버헤드가 포함되지 않습니다. 이는 L2 오버헤드가 포함된 XR 인터페이스 **config** 및 **show** 명령과는 다릅니다. 하위 인터페이스의 MTU는 2018바이트 (2014바이트의 기본 인터페이스에서 상속)이지만, LDP는 MTU를 2000바이트로 신호를 보냈습니다. 그 결과 L2 헤더에서 18바이트(이더넷 헤더 14바이트 + 1dot1q 태그 4바이트)를 뺍니다.

MTU 불일치를 해결하기 위해 각 디바이스가 첨부 회로의 MTU 값을 계산하는 방법을 이해하는 것이 중요합니다. 이는 공급업체, 플랫폼, 소프트웨어 버전 및 컨피그레이션과 같은 매개변수에 따라 달라집니다.

EVC(ASR9000)

Cisco ASR 9000 Series Aggregation Services Router는 L2VPN L2 인터페이스 및 하위 인터페이스에서 유연한 VLAN 일치 허용하는 EVC 인프라 모델을 사용합니다.

EVC L2VPN L2 인터페이스에는 다음과 같은 특성이 있습니다.

- encapsulation 명령으로 하나 이상의 태그를 구성할 수 있습니다.
- 기본적으로 캡슐화 명령만 사용하여 태그는 PW를 통해 보존되고 전송됩니다. 따라서 비 EVC 플랫폼에서 태그를 삭제해야 하므로 기본적으로 태그를 제거할 필요가 없습니다.
- 수신 태그를 팝업하거나 수신 프레임 위에 추가 태그를 푸시하려면 rewrite 명령을 사용합니다.

하위 인터페이스 MTU를 계산하려면 기본 인터페이스 MTU(기본값 또는 기본 인터페이스 아래에 수동으로 구성된 인터페이스)를 사용하고, encapsulation 명령으로 구성된 각 VLAN 태그에 4바이트를 추가합니다. 자세한 내용은 [특정 EFP 캡슐화 명령을 참조하십시오](#).

하위 인터페이스 아래에 mtu 명령이 있는 경우 계산된 MTU보다 낮은 경우에만 적용됩니다. rewrite 명령은 하위 인터페이스의 MTU에 영향을 주지 않습니다.

예를 들면 다음과 같습니다.

```
RP/0/RP0/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)
```

이 예에서 기본 인터페이스의 MTU는 2014바이트입니다. 하위 인터페이스 아래에 두 개의 태그가 구성되어 있으므로 8바이트를 추가합니다.

하위 인터페이스에서 **mtu 2026**바이트를 구성하면 기본 인터페이스(2022)에서 상속된 하위 인터페이스의 MTU보다 크기 때문에 적용되지 않습니다. 따라서 2022바이트보다 작은 하위 인터페이스 MTU만 구성할 수 있습니다.

이 하위 인터페이스 MTU를 기반으로 네이버에 시그널링되는 MPLS LDP 페이로드의 MTU를 계산하고, 원격 L2VPN PE에서 계산한 페이로드와 동일한지 확인합니다. 여기서 **rewrite 명령**이 실행됩니다.

MPLS LDP 페이로드의 MTU를 계산하려면 하위 인터페이스의 MTU를 가져온 다음 다음을 수행합니다.

1. 이더넷 헤더에 대해 14바이트를 뺍니다.
2. 하위 인터페이스 아래에 구성된 rewrite 명령에서 팝된 각 태그에 대해 4바이트를 뺍니다.
3. 하위 인터페이스 아래에 구성된 rewrite 명령에서 푸시된 각 태그에 대해 4바이트를 추가합니다.

이는 gig 0/1/0/3.201의 QinQ 컨피그레이션과 동일한 예입니다.

```
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)
```

이는 MPLS LDP 페이로드의 MTU에 대한 계산입니다.

1. 하위 인터페이스 MTU의 MTU 값: 2022바이트
2. 이더넷 헤더 14바이트 빼기: $2022 - 14 = 2008$ 바이트
3. 에서 팝업된 각 태그에 대해 4바이트 빼기 **재작성 명령**: $2008 - 4 * 2 = 2000$

원격측에서 2000바이트의 MPLS LDP 페이로드를 광고하는지 확인합니다. 그렇지 않으면 로컬 또

는 원격 연결 회로(AC) MTU 크기가 일치하도록 조정합니다.

```
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
```

특정 EFP(Ethernet Flow Point) 캡슐화 명령

이러한 캡슐화는 일치하는 제로 태그로 계산되므로 하위 인터페이스 MTU를 증가시키지 않습니다.

- 캡슐화 태그 없음
- 캡슐화 기본값

이러한 캡슐화 수정자는 하위 인터페이스 MTU를 계산하는 데 필요한 태그 수에 영향을 주지 않습니다.

- 네이티브
- 페이로드 이더 타입
- 정확하
- 코스
- 인그레스 source-mac 또는 인그레스 destination-mac

캡슐화 [dot1q|dot1ad] 우선 순위 태그가 지정된 경우 단일 태그와 일치하는 것으로 간주됩니다.

가장 안쪽 태그 일치로 사용되는 'any' 키워드는 하위 인터페이스 MTU를 증가시키지 않습니다.

- 캡슐화 dot1q any는 하위 인터페이스 MTU를 증가시키지 않습니다.
- 캡슐화 dot1ad 10 dot1q any는 하나의 태그로 간주되며, 하위 인터페이스 MTU가 4바이트 증가합니다.
- 캡슐화 dot1ad any dot1q 7은 두 개의 태그로 간주되며, 하위 인터페이스 MTU가 8바이트 증가합니다.

VLAN-ID의 범위는 하위 인터페이스 MTU를 증가시킵니다.

• 캡슐화 dot1q 10-100은 하나의 태그로 간주되며, 하위 인터페이스 MTU가 4바이트 증가합니다. 구분 매칭인 EFP의 캡슐화 MTU 오버헤드는 최상위 요소의 MTU로 처리됩니다.

- 캡슐화 dot1q 10-100, 10-100 범위가 가장 높은 요소이기 때문에 태그가 지정되지 않은 경우 하나의 태그로 간주됩니다.

비 EVC(XR 12000 및 CRS)

Cisco XR 12000 Series 라우터 및 CRS(Carrier Routing System)와 같은 라우터는 하위 인터페이스에서 VLAN 일치를 위해 기존 컨피그레이션을 사용합니다. 이러한 특성은 EVC 모델을 따르지 않는 CRS 및 XR 12000 라우터의 L2VPN L2 인터페이스에 적용됩니다.

- 비 EVC 플랫폼에서 들어오는 dot1q 또는 dot1ad 태그는 L2 전송 하위 인터페이스에서 수신될 때 자동으로 제거됩니다.

- 신호를 보낼 MPLS LDP의 페이로드 크기를 계산할 때는 **show interface** 명령에 나와 있는 것처럼 하위 인터페이스의 MTU에서 태그 크기를 뺍니다.
- 이는 라우트드 하위 인터페이스의 경우와 유사합니다.
- 하위 인터페이스는 기본 인터페이스에서 MTU를 상속합니다. 하위 인터페이스의 MTU를 계산하기 위해 각 태그의 4바이트를 기본 인터페이스의 MTU에 추가합니다. 예를 들어, QinQ 하위 인터페이스에 dot1q 태그가 2개 있는 경우, 기본적으로 하위 인터페이스의 MTU는 기본 인터페이스의 MTU보다 8바이트 더 큼니다.
- 또한 하위 인터페이스 아래에서 **mtu** 명령을 사용할 수 있지만, 이 명령은 기본 인터페이스의 MTU에서 상속되는 하위 인터페이스의 MTU를 줄이는 데만 사용됩니다.

이러한 특징을 보여주는 몇 가지 예를 소개한다.

이 예에서는 비 EVC 하위 인터페이스가 어떻게 구성되었는지 보여줍니다.

```
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#
```

비 EVC 플랫폼은 EVC 플랫폼(ASR9000)의 **캡슐화 및 재작성 명령** 대신 **dot1q vlan** 또는 **dot1ad vlan** 명령을 사용합니다.

기본 또는 하위 인터페이스에서 MTU를 명시적으로 구성하지 않으면 기본적으로 1500바이트 L3 패킷이 수신될 수 있습니다.

```
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#
```

하위 인터페이스 MTU는 기본 인터페이스 MTU(1514)에서 계산되며, 각 dot1q 태그에 4바이트를 추가합니다. 하위 인터페이스에 **dot1q vlan 201** 명령으로 구성된 태그가 하나 있으므로 MTU 1518바이트에 대해 1514에 4바이트를 추가합니다.

MPLS LDP의 해당 페이로드 MTU는 1500바이트입니다. 이더넷 헤더의 14바이트가 계산되지 않고, 비 EVC 플랫폼에서 PW를 초과할 때 dot1q 태그 하나가 자동으로 팝업되기 때문입니다.

```
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
```

주 인터페이스의 MTU를 2014바이트로 늘리면 하위 인터페이스의 MTU가 그에 따라 증가합니다.

```
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
```

따라서 MPLS LDP MTU를 계산하기 위해 이더넷 헤더 14바이트를 빼고, 하위 인터페이스 아래에 구성된 각 태그에 4바이트를 더합니다.

자동 이더넷 인터페이스 드라이버 MTU 및 MRU 구성

이더넷 인터페이스에서 인터페이스 드라이버는 인터페이스 MTU 컨피그레이션을 기반으로 하는 MTU 및 MRU로 구성됩니다.

이더넷 인터페이스 드라이버에 구성된 MTU 및 MRU는 **show controller <interface> all** 명령으로 확인할 수 있습니다.

Cisco IOS XR Release 5.1.1 이전 릴리스에서는 인터페이스의 Cisco IOS XR MTU 컨피그레이션에 따라 이더넷 인터페이스 드라이버의 MTU 및 MRU가 자동으로 구성되었습니다.

이더넷 드라이버에 구성된 MTU/MRU는 2개의 이더넷 태그 및 CRC 필드를 추가하기 위해 구성된 MTU + 12바이트를 기반으로 했습니다. 하위 인터페이스에 구성된 VLAN 태그가 있는지 여부에 관계없이 12바이트가 이더넷 드라이버 MTU/MRU에 추가되었습니다.

Cisco IOS XR Release 5.1.1 이전의 모든 Cisco IOS XR 버전 및 ASR 9000 인터페이스의 기본 MTU 1514의 예는 다음과 같습니다.

```
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 Release 5.1.1 이상에서는 이더넷 인터페이스 드라이버에 사용되는 MTU 및 MRU가 변경되었으며 이제 하위 인터페이스에 구성된 VLAN 태그 수를 기준으로 합니다.

하위 인터페이스에 구성된 VLAN 태그가 없는 경우 드라이버 MTU/MRU는 인터페이스의 구성된 MTU + 4 CRC 바이트(예: 1514 + 4 = 1518 바이트)와 같습니다.

하위 인터페이스에 하나의 VLAN이 구성된 경우 드라이버 MTU/MRU는 구성된 MTU + 8바이트(1태그 + CRC)와 같습니다(예: 1514 + 8 = 1522바이트).

임의의 하위 인터페이스에 2개의 VLAN 태그가 구성된 경우 드라이버 MTU/MRU는 구성된 MTU + 12바이트(2개 태그 + CRC)와 같습니다(예: 1514 + 12 = 1526바이트)

any 키워드가 포함된 QinQ가 두 번째 do1q 태그에 대해 구성된 경우 드라이버 MTU/MRU는 구성된 MTU + 8바이트(1 태그 + CRC)와 같습니다(예: 1514 + 8 = 1522바이트).

다음 예에서는 ASR 9000의 Cisco IOS XR 릴리스 5.1.1 이상에서 동작을 표시합니다.

```
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>

대부분의 경우 릴리스 5.1.1 이상에서 이 동작이 변경되어도 인터페이스의 MTU 컨피그레이션을 변경할 필요가 없습니다.

이러한 동작 변경은 단일 VLAN 태그로 구성된 하위 인터페이스의 경우 문제를 일으킬 수 있지만, 두 개의 VLAN 태그로 패킷을 수신합니다. 이 경우 수신된 패킷이 이더넷 인터페이스 드라이버의 MRU를 초과할 수 있습니다. 이러한 조건을 제거하기 위해 인터페이스 MTU를 4바이트 늘리거나 2개의 VLAN 태그로 구성된 하위 인터페이스를 사용할 수 있습니다.

릴리스 5.1.1의 자동 이더넷 인터페이스 드라이버 MTU 및 MRU 구성은 CRS 및 ASR 9000 라우터에 대해 동일합니다. 그러나 릴리스 5.1.1을 실행하는 CRS 라우터에서는 **show controller** 출력에 표시된 MTU 및 MRU 값에 4바이트 CRC가 포함되지 않습니다. 보고된 방식의 동작은 CRS와 ASR9000 간에 동일하지 않습니다.

```
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)
```

MTU 및 MRU가 ASR 9000의 show controller 출력에 표시되는 방식은 나중에 변경되어 CRC 4바이트가 표시된 MTU/MRU 값에 포함되지 않습니다. 이러한 향후 변경은 Cisco 버그 ID CSCuo93379를 사용하여 추적할 수 있습니다.

릴리스 5.1.1 이전 릴리스에서 릴리스 5.1.1 이상으로 업그레이드할 때 컨피그레이션 변환

- 기본 MTU:

릴리스 5.1.1 이전의 릴리스에서 하위 인터페이스가 없고 mtu 명령이 없는 기본 인터페이스가 있는 경우:

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

이 인터페이스는 dot1q 또는 QinQ 프레임을 전송하므로 릴리스 5.1.1 이상에서 MTU를 "mtu 1522"로 수동으로 구성해야 합니다.

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

이 컨피그레이션을 사용하면 이전 릴리스에서와 같이 QinQ 프레임을 전송할 수 있습니다. QinQ가 아닌 dot1q만 전송하려면 MTU 값을 1518로 구성할 수 있습니다.

dot1q 또는 QinQ에 대해 구성된 하위 인터페이스가 있지만 "any" 키워드와 함께 2개의 명시적 태그가 있는 QinQ 하위 인터페이스가 Release 5.1.1 이전 릴리스에서 구성되지 않은 경우:

```
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
!
```

Release 5.1.1 이상에서는 이 컨피그레이션을 통해 하나의 태그로 프레임을 전송할 수 있으므로 QinQ 프레임을 전송하려면 MTU를 4바이트 수동으로 늘려야 합니다.

```
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
!
```

2개의 명시적 태그("any" 키워드를 사용하지 않음)가 포함된 QinQ 하위 인터페이스가 구성된 경우, Release 5.1.1 이상으로 업그레이드할 때 MTU 컨피그레이션을 수정할 필요가 없습니다.

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

L2 전송 하위 인터페이스가 없고 L3 라우티드 인터페이스만 있는 경우 MTU 컨피그레이션이 양쪽에서 일치하고 전송되는 MTU보다 큰 프레임이 없을 것으로 예상됩니다. 릴리스 5.1.1 이상으로 업그레이드할 경우 MTU 컨피그레이션을 업데이트할 필요가 없습니다.

• 릴리스 5.1.1 이전 릴리스의 기본이 아닌 MTU:

마찬가지로, 릴리스 5.1.1 이전 릴리스에서 기본이 아닌 MTU가 구성되었고 하위 인터페이스가 구성되지 않았으며 dot1q 또는 QinQ 프레임을 전송해야 하는 경우 릴리스 5.1.1 이상으로 업그레이드할 때 구성된 MTU 값을 8바이트 늘려야 합니다.

릴리스 5.1.1 이전 릴리스:

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

릴리스 5.1.1 이상으로 업그레이드할 경우 MTU를 8바이트 수동으로 늘려야 합니다.

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

dot1q 하위 인터페이스가 있고 QinQ 하위 인터페이스가 없거나 두 번째 dot1q 태그에 대해 any 키워드가 포함된 QinQ 하위 인터페이스가 없는 경우에도 구성된 MTU 값이 4바이트 증가해야 합니다

릴리스 5.1.1 이전 릴리스:

```
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
!
```

릴리스 5.1.1 이상:

```
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
!
```

2개의 명시적 태그("any" 키워드를 사용하지 않음)가 포함된 QinQ 하위 인터페이스가 구성된 경우, 릴리스 5.1.1 이상으로 업그레이드할 때 MTU 컨피그레이션을 수정할 필요가 없습니다.

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

L2 전송 하위 인터페이스가 없고 L3 라우티드 인터페이스만 있는 경우 MTU 컨피그레이션이 양쪽에서 일치하고 전송되는 MTU보다 큰 프레임이 없을 것으로 예상됩니다. 릴리스 5.1.1 이상으로 업그레이드할 경우 MTU 컨피그레이션을 업데이트할 필요가 없습니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.