

Comportamento MTU no Cisco IOS XR e no Roteadores do Cisco IOS

Índice

[Introdução](#)

[Informações de Apoio](#)

[Configurar](#)

[Comparação do Cisco IOS e do Software Cisco IOS XR](#)

[Relações L3 roteados](#)

[MTU padrão](#)

[MTU não-padrão](#)

[Subinterfaces L3 roteados](#)

[Relação L2VPN L2](#)

[EVC \(ASR9000\)](#)

[Não EVC \(XR 12000 e CR\)](#)

[Direcionador automático MTU da interface Ethernet e configuração MRU](#)

[Converta a configuração quando você promove de uma liberação mais cedo do que liberam 5.1.1 para liberar 5.1.1 ou mais atrasado](#)

Introdução

Este documento descreve comportamentos da unidade de transmissão máxima (MTU) no Roteadores do [®] XR do Cisco IOS e compara aqueles comportamentos ao Roteadores do Cisco IOS. Igualmente discutem MTU em relações roteados da camada 3 (L3) e as relações L2 da camada 2 VPN (L2VPN) que usam os modelos conexão virtual dos Ethernet (EVC) e NON-EVC. Este documento igualmente descreve mudanças importantes a como o direcionador MTU da interface Ethernet e o Maximum Receive Unit (MRU) são configurados automaticamente na liberação 5.1.1 e mais atrasado.

Informações de Apoio

Em trabalhos em rede do computador, o MTU de um protocolo de comunicações de uma camada define o tamanho, nos bytes, da unidade de dados de protocolo a maior que a camada é permitida transmitir sobre uma relação. Um parâmetro MTU é associado com cada relação, camada, e protocolo.

As características MTU no Software Cisco IOS XR são:

- **A configuração** e os **comandos show MTU**, no L2 e no L3, incluem o tamanho de cabeçalho de sua camada de correspondência. Por exemplo, o **comando mtu** que configura o L2 MTU

inclui 14 bytes para uma interface Ethernet (sem dot1q), ou 4 bytes para o Point-to-Point Protocol (PPP) ou o High-Level Data Link Control (HDLC). O comando **mtu do IPv4** inclui 20 bytes do encabeçamento do IPv4.

- O MTU de uma camada superior deve caber dentro do payload da camada mais baixa. Por exemplo, se a interface MTU de uma interface Ethernet non-dot1q é o padrão de 1514 bytes, a seguir os protocolos de camada mais elevada tais como o Multiprotocol Label Switching (MPLS) podem ter um MTU máximo de 1500 bytes nessa relação. Isto significa que você pode caber somente um quadro de 1500 bytes MPLS (que inclui etiquetas) dentro do frame da Ethernet. Você não pode configurar 1508 um byte MPLS MTU nessa relação se você quer permitir duas etiquetas MPLS sobre um pacote IPv4 de 1500 bytes. A fim transmitir um quadro de 1508 bytes MPLS em uma interface Ethernet, a interface MTU deve ser aumentada a 1522, ou a um valor mais alto, a fim assegurar-se de que o payload da relação L2 seja grande bastante levar o quadro MPLS.
- No Cisco IOS Software clássico (não o Software Cisco IOS XR), o **comando mtu da relação** configura o tamanho de virulência L2, mas não inclui o encabeçamento L2. Isto é diferente do Software Cisco IOS XR que inclui as despesas gerais L2 e L3 no **comando mtu da relação**. Os comandos mtu L3, como no caso do **comando mtu do IPv4**, configuram o tamanho máximo do pacote desse protocolo que inclui o encabeçamento L3. Isto é similar ao exemplo do Software Cisco IOS XR.
- A interface padrão MTU no Software Cisco IOS XR deve permitir o transporte de um pacote de 1500 bytes L3. Consequentemente, o MTU padrão é 1514 bytes para uma interface Ethernet principal e 1504 bytes para uma interface serial.

O restante deste documento ilustra características MTU, compara o comportamento do Cisco IOS e do Software Cisco IOS XR, e dá exemplos para estes tipos de relações:

- Relações L3 roteados
- Subinterfaces L3 roteados
- Relações L2VPN L2

Configurar

Nota: Use a [Command Lookup Tool](#) ([somente clientes registrados](#)) para obter mais informações sobre os comandos usados nesta seção.

Nota: [A ferramenta Output Interpreter](#) ([clientes registrados somente](#)) apoia determinados comandos de exibição. Use a ferramenta Output Interpreter a fim ver uma análise do emissor de comando de execução.

Comparação do Cisco IOS e do Software Cisco IOS XR

Esta seção compara o comportamento do Cisco IOS e do Software Cisco IOS XR com referência às características MTU.

No Cisco IOS Software, o **comando mtu** e os **comandos show** correspondentes não incluem o encabeçamento L2. Use o **comando mtu** a fim configurar o payload L2 ao tamanho máximo para os pacotes L3, incluindo o encabeçamento L3.

Isto é diferente do Software Cisco IOS XR, onde o **comando mtu** inclui o encabeçamento L2 (14 bytes para Ethernet ou 4 bytes para o PPP/HDLC).

Se um roteador do Cisco IOS é configurado com **MTU x** e conectado a um roteador do Cisco IOS XR, a seguir a interface correspondente no roteador do Cisco IOS XR deve ser configurada com o **MTU x+14** para interfaces Ethernet, ou o **MTU x+4** para interfaces serial.

O Cisco IOS e o Software Cisco IOS XR têm o mesmo significado para o **MTU do IPv4**, o **MTU do IPv6** e os **comandos mpls mtu**; devem ser configurados com os mesmos valores.

Em consequência, esta é a configuração no Cisco IOS Software em uma interface Ethernet:

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

A configuração correspondente no vizinho do Software Cisco IOS XR é:

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

Relações L3 roteados

Os valores MTU devem ser os mesmos em todos os dispositivos conectados a uma rede L2. Se não, estes sintomas puderam ser relatados:

- As adjacências do Intermediate System-to-Intermediate System (IS-IS) não vêm acima. À revelia, o IS-IS usa o preenchimento de hello; consequentemente, os hellos puderam ser caracterizados como gigantes e puderam ser deixados cair quando um roteador tem um valor MTU que fosse mais baixo do que os valores no outro Roteadores.
- As adjacências do Open Shortest Path First (OSPF) obtêm coladas em Exstart ou no estado de intercâmbio, porque os grandes pacotes do Database Descriptor (DBD) puderam ser caracterizados como gigantes e puderam ser deixados cair. Quando os pacotes são recebidos em um roteador com um valor do MTU inferior, os bases de dados não estão sincronizados.
- O tráfego de dados está caracterizado como gigantes e deixado cair quando está recebido em um dispositivo com um valor MTU que seja mais baixo do que esse no dispositivo transmissor.
- Há um throughput baixo quando os grandes pacotes obtêm deixados cair. Em caso do Path MTU Discovery, a sessão de TCP pode recuperar quando os grandes pacotes são deixados cair, mas este afeta a taxa de transferência.

MTU padrão

Esta seção analisa o MTU padrão de uma interface roteada quando o **comando mtu** não é configurado:

```
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#
```

Neste exemplo, a interface MTU do padrão L2 é 1514 bytes e inclui 14 bytes do cabeçalho de Ethernet. Os 14 bytes são explicados pelos bytes 6 do endereço MAC de destino, dos bytes 6 do endereço MAC de origem, e dos 2 bytes do tipo ou do comprimento. Isto não inclui o preâmbulo, o delimitador do quadro, os 4 bytes da sequência de verificação de frame (FCS), e a diferença do inter-quadro. Para um quadro PPP ou HDLC, 4 bytes do encabeçamento L2 são esclarecidos; assim a interface padrão MTU é 1504 bytes.

Os protocolos da criança L3 herdam seu MTU do payload do pai MTU. Quando você subtrai 14 bytes de um encabeçamento L2 de um L2 MTU de 1514 bytes, você tem um payload L2 de 1500 bytes. Este transforma-se o MTU para os protocolos L3. O IPv4, o IPv6, o MPLS, e o serviço de rede sem conexão (CLNS) herdam this1500 o byte MTU. Em consequência, uma interface Ethernet do Cisco IOS XR, à revelia, pode transportar um pacote de 1500 bytes L3 que seja o mesmo que o default em uma interface Ethernet do Cisco IOS.

MTU não-padrão

Esta seção mostra como configurar um **MTU dos mpls de 1508** a fim enviar um pacote IPv4 de 1500 bytes com as duas etiquetas MPLS de 4 bytes cada um, sobre o pacote:

```
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#
```

Embora o comando **MTU 1508 dos mpls** seja comprometido, não é aplicado, porque o MPLS ainda tem um MTU de 1500 bytes no **comando show**. Isto é porque os protocolos da criança L3 não podem ter um MTU maior do que o payload de sua relação do pai L2.

A fim permitir duas etiquetas sobre um pacote do IP de byte 1500, você deve:

- Configurar uma interface MTU L2 de 1522 bytes, de modo que todos os protocolos da criança (que incluem o MPLS) herdem um MTU de 1508 bytes (1522 - 14 = 1508).
- Reduza o MTU dos protocolos L3 a 1500 bytes, de modo que somente o MPLS seja permitido exceder 1500 bytes.

```
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 ip 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#

Esta configuração deixa-o enviar pacotes do IPv4 e do IPv6 de 1500 bytes e pacotes de MPLS de 1508 bytes (um pacote de bytes 1500 com as duas etiquetas na parte superior).

Subinterfaces L3 roteados

Estas características aplicam-se distribuíram as subinterfaces L3.

Uma secundário-relação roteado MTU herda o MTU de sua interface principal do pai; adicionar 4 bytes para cada etiqueta VLAN configurada na secundário-relação. Assim, há 4 bytes para uma secundário-relação do dot1q e 8 bytes para um IEEE 802.1Q que escava um túnel a secundário-relação (de QinQ).

Em consequência, os pacotes L3 do mesmo tamanho podem ser enviados na interface principal e na secundário-relação.

O comando **mtu** pode ser configurado sob a secundário-relação, mas é aplicada somente se é mais baixa ou igual ao MTU que está herdado da interface principal.

Este é um exemplo onde o MTU da interface principal seja 2000 bytes:

```
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#
```

Nos comandos show, o MTU da secundário-relação é 2004; adicionar 4 bytes ao MTU da interface principal porque há uma etiqueta do dot1q configurada sob a secundário-relação.

Contudo, o MTU dos pacotes do IPv4 e do IPv6 é ainda o mesmo que aquele da interface principal (1986). Isto é porque o MTU dos protocolos L3 é computado agora como: $2004 - 14 - 4 = 1986$.

O comando mtu pode ser configurado sob a secundário-relação, mas o MTU configurado é aplicado somente se é mais baixo ou igual ao MTU que está herdado da interface principal (4 bytes maior do que o MTU da interface principal).

Quando o MTU da secundário-relação que é maior do que o MTU herdado, ele não for aplicado, como mostrado aqui:

```
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#
```

Assim, você pode usar somente o **comando mtu** a fim abaixar o valor MTU herdado da interface principal.

Similarmente, você pode igualmente usar os comandos mtu dos protocolos L3 (IPv4, IPv6, MPLS) a fim abaixar o valor do L3 MTU herdado do secundário-relação L2. L3 o protocolo MTU não toma o efeito quando é configurado a um valor que não caiba no payload do L2 MTU.

Relação L2VPN L2

O MTU para um L2VPN é importante porque o protocolo de distribuição de rótulo (LDP) não traz um pseudowire (picowatt) acima de quando os MTU nos circuitos do acessório em cada lado de um picowatt não são os mesmos.

Está aqui um **comando show** que ilustre que um L2VPN picowatt fica para baixo quando há uma má combinação MTU:

```
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#

```

Neste exemplo, note que as pontas de provedor MPLS L2VPN (PE) em cada lado devem sinalizar o mesmo valor MTU a fim trazer acima o picowatt.

O MTU sinalizado por MPLS LDP não inclui as despesas gerais L2. Isto é diferente da **configuração** e dos **comandos show** da relação XR que incluem as despesas gerais L2. O MTU na secundário-relação é 2018 bytes (como herdado da interface principal de 2014 bytes), mas o LDP sinalizou um MTU de 2000 bytes. Em consequência, subtraí 18 bytes (14 bytes do cabeçalho de Ethernet + 4 bytes de 1 etiqueta do dot1q) do encabeçamento L2.

É importante compreender como cada dispositivo computa os valores MTU dos circuitos do acessório a fim fixar más combinações MTU. Isto depende em cima dos parâmetros tais como o vendedor, a plataforma, a versão de software, e a configuração.

EVC (ASR9000)

A agregação do 9000 Series de Cisco ASR presta serviços de manutenção ao roteador usa o modelo da infraestrutura EVC, que permite o VLAN flexível que combina em relações e em subinterfaces L2VPN L2.

As relações EVC L2VPN L2 têm estas características:

- Permitem a configuração de umas ou várias etiquetas com o **comando encapsulation**.
- À revelia e com apenas o **comando encapsulation**, as etiquetas são preservadas e transportadas sobre PWs. Em consequência, você não precisa de descascar à revelia etiquetas, enquanto você precisa de fazer nas Plataformas NON-EVC.
- Use o comando da **reescrita** quando você decide estalar as etiquetas entrantes ou empurrar algumas etiquetas adicionais sobre o frame de entrada.

A fim computar a secundário-relação MTU, tomar a interface principal MTU (o padrão ou esse configurado manualmente sob a interface principal), e adicionar 4 bytes para cada etiqueta VLAN configurada com o **comando encapsulation**. Veja [comandos encapsulation específicos EFP](#).

Quando há um **comando mtu** sob a secundário-relação, toma o efeito somente se é mais baixa do que o MTU computado. O comando da **reescrita** não influencia o MTU da secundário-relação.

Aqui está um exemplo:

```
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)
```

Neste exemplo, o MTU na interface principal é 2014 bytes; adicionar 8 bytes porque há duas etiquetas configuradas sob a secundário-relação.

Se você configura o **MTU 2026** bytes sob a secundário-relação, não é aplicada porque é maior do que o MTU da secundário-relação herdada da interface principal (2022). Em consequência, você pode configurar somente uma secundário-relação MTU mais baixo de 2022 bytes.

Baseado nesta secundário-relação MTU, compute o MTU do payload MPLS LDP que é sinalizado ao vizinho, e certifique-se de que é idêntico a esse computado pelo L2VPN remoto PE. Isto é o lugar aonde o comando da **reescrita** entra o jogo.

A fim computar o MTU do payload MPLS LDP, tome o MTU da secundário-relação, então:

1. Subtraia 14 bytes para o cabeçalho de Ethernet.
2. Subtraia 4 bytes para cada etiqueta estalada no comando da **reescrita** configurado sob a secundário-relação.
3. Adicionar 4 bytes para cada etiqueta empurrada o comando da **reescrita** configurado sob a secundário-relação.

Este é o mesmo exemplo com a configuração de QinQ na atuação 0/1/0/3.201:

```
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)
```

Esta é a computação para o MTU do payload MPLS LDP:

1. Valor MTU da secundário-relação MTU: 2022 bytes
2. Subtraia 14 bytes do cabeçalho de Ethernet: $2022 - 14 = 2008$ bytes
3. Subtraia 4 bytes para cada etiqueta estalada no comando da **reescrita**: $2008 - 4 * 2 = 2000$

Assegure-se de que o lado remoto anuncie um payload MPLS LDP de 2000 bytes. Se não, ajuste o local ou tamanho do MTU remoto do circuito do acessório (AC) assim que combina.

```
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
```

Os Ethernet específicos fluem comandos encapsulation do ponto (EFP)

Estes encapsulamentos contam como etiquetas zero de harmonização, assim que não aumentam a secundário-relação MTU:

- **sem etiqueta do encapsulamento**
- **padrão do encapsulamento**

Estes modificadores do encapsulamento não afetam o número de etiquetas exigidas a fim computar a secundário-relação MTU:

- nativo
- payload-Ethertype
- exato
- cos
- ingresso fonte-MAC ou ingresso destino-MAC

o encapsulamento [dot1q|dot1ad] prioridade-etiquetou contagens como a harmonização de uma única etiqueta.

“Nenhuma” palavra-chave usada como o fósforo de etiqueta mais íntimo não aumenta a secundário-relação MTU.

- **o dot1q do encapsulamento** não aumenta a secundário-relação MTU.
- **o dot1q do encapsulamento dot1ad 10** é explicado como uma etiqueta; aumenta a secundário-relação MTU por 4 bytes.
- **o encapsulamento dot1ad todo o dot1q 7** é explicado como duas etiquetas; aumenta a secundário-relação MTU por 8 bytes.

As escalas dos ID de VLAN incrementam a secundário-relação MTU:

- **o dot1q 10-100 do encapsulamento** é explicado como uma etiqueta; aumenta a secundário-relação MTU por 4 bytes.

As despesas gerais do encapsulamento MTU de um EFP que seja um fósforo disjuntivo são tratadas como o MTU de seu elemento mais alto.

- **o dot1q 10-100 do encapsulamento, sem etiqueta** é explicado porque uma etiqueta porque a

escala 10 -100 é o elemento o mais alto.

Não EVC (XR 12000 e CR)

O Roteadores como o Roteador Cisco XR série 12000 e o sistema de roteamento do portador (CR) usa a configuração tradicional para o VLAN que combina em subinterfaces. Estas características aplicam-se às relações L2VPN L2 em CR e nos 12000 Router XR que não seguem o modelo EVC:

- Nas Plataformas NON-EVC, o dot1q entrante ou as etiquetas dot1ad são descascados automaticamente quando é recebido em uma secundário-relação do transporte L2.
- Quando você está computando o tamanho de virulência para MPLS LDP para sinalizar, subtraia o tamanho das etiquetas do MTU da secundário-relação, como visto no **comando show interface**.
- Isto é similar ao exemplo de uma secundário-relação roteado.
- A secundário-relação herda seu MTU da interface principal; adicionar os 4 bytes para cada etiqueta ao MTU da interface principal a fim computar o MTU da secundário-relação. Por exemplo, se uma secundário-relação de QinQ tem 2 etiquetas do dot1q, a secundário-relação, à revelia, tem um MTU que seja 8 bytes maior do que o MTU da interface principal.
- Você pode igualmente usar o **comando mtu** sob a secundário-relação, mas é usada para reduzir somente o MTU da secundário-relação, que é herdada do MTU da interface principal.

Estão aqui diversos exemplos que ilustram estas características.

Este exemplo mostra como uma secundário-relação NON-EVC é configurada:

```
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#
```

As Plataformas NON-EVC usam o **dot1q vlan** ou **comandos vlan dot1ad** em vez do **encapsulamento e reescrevem** comandos das Plataformas EVC (ASR9000).

Se você não configura um MTU explicitamente no cano principal ou na secundário-relação, a seguir um pacote de 1500 bytes L3 pode ser recebido à revelia:

```
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#
```

A secundário-relação MTU é computada da interface principal MTU (1514); adicionar 4 bytes para cada etiqueta do dot1q. Porque há uma etiqueta configurada na secundário-relação com o **dot1q 201 vlan** comanda, adiciona 4 bytes a 1514 para um MTU de 1518 bytes.

O payload correspondente MTU em MPLS LDP é 1500 bytes, desde que os 14 bytes do cabeçalho de Ethernet não estão contados e a uma etiqueta do dot1q está estalada automaticamente pela plataforma NON-EVC quando vai sobre o picowatt:

```
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
```

Se você aumenta o MTU da interface principal a 2014 bytes, o MTU da secundário-relação está aumentado em conformidade:

```
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
```

Assim, a fim computar o MPLS LDP MTU, subtrair 14 bytes do cabeçalho de Ethernet, e adicionar 4 bytes para cada etiqueta configurada sob a secundário-relação.

Direcionador automático MTU da interface Ethernet e configuração MRU

Em interfaces Ethernet o direcionador da relação é configurado com um MTU e um MRU que seja baseado na configuração da interface MTU.

O MTU e o MRU configurados no direcionador da interface Ethernet podem ser vistos com o comando **all do <interface> do controlador da mostra**.

Nas liberações mais cedo do que o Cisco IOS XR libere 5.1.1, o MTU e o MRU no direcionador da interface Ethernet foram configurados automaticamente com base na configuração do Cisco IOS XR MTU na relação.

O MTU/MRU configurado na unidade Ethernet foi baseado simplesmente no MTU+ configurado 12 bytes para a adição de 2 etiquetas dos Ethernet e do campo de CRC. Os 12 bytes foram adicionados à unidade Ethernet MTU/MRU apesar de se havia alguma etiqueta VLAN configurada nas subinterfaces.

Um exemplo com todas as versões do Cisco IOS XR mais cedo do que a liberação 5.1.1 do Cisco IOS XR e um MTU padrão de 1514 em uma relação ASR 9000 é mostrado aqui:

```
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>
```

No Cisco IOS XR libere 5.1.1 e mais atrasado, o MTU e o MRU que é usado no direcionador da interface Ethernet mudou e é baseado agora no número de etiquetas VLAN que são configuradas em algumas das subinterfaces.

Se nenhuma etiqueta VLAN é configurada em quaisquer subinterfaces, o direcionador MTU/MRU iguala o MTU configurado na relação + 4 bytes CRC, por exemplo $1514 + 4 = 1518$ bytes.

Se um VLAN é configurado em quaisquer subinterfaces, o direcionador MTU/MRU iguala o MTU+ configurado 8 bytes (1 etiqueta + CRC), por exemplo $1514 + 8 = 1522$ bytes.

Se duas etiquetas VLAN são configuradas em quaisquer subinterfaces, o direcionador MTU/MRU iguala o MTU+ configurado 12 bytes (2 etiquetas + CRC), por exemplo $1514 + 12 = 1526$ bytes

Se QinQ com a **qualquer** palavra-chave é configurado para a etiqueta second-do1q o direcionador MTU/MRU iguala o MTU+ configurado 8 bytes (1 etiqueta + CRC), por exemplo $1514 + 8 = 1522$ bytes.

Estes exemplos indicam o comportamento na liberação 5.1.1 do Cisco IOS XR e mais tarde em um ASR 9000:

```
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>
```

Na maioria das situações esta mudança do comportamento na liberação 5.1.1 e mais atrasado não deve exigir nenhuma mudanças à configuração do MTU na relação.

Esta mudança do comportamento pode causar problemas no caso de uma secundário-relação configurada com uma única etiqueta VLAN, mas recebe pacotes com as duas etiquetas VLAN. Nessa situação os pacotes recebidos podem exceder o MRU no direcionador da interface

Ethernet. A fim eliminar essa circunstância, a interface MTU pode ser aumentada por 4 bytes ou pela secundário-relação configurada com as duas etiquetas VLAN.

O direcionador automático MTU da interface Ethernet e a configuração MRU no comportamento da liberação 5.1.1 são o mesmo para 9000 Router CR e ASR. Mas um roteador CR que execute a liberação 5.1.1 não inclui 4 o byte CRC no valor MTU e MRU indicado nas **saídas do controlador da mostra**. O comportamento de como se relata não é o mesmo entre CR e ASR9000.

```
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)
```

A maneira o MTU e o MRU é indicada nas saídas do controlador da mostra no ASR 9000 mudará no futuro de modo que os 4 bytes do CRC não sejam incluídos no valor MTU/MRU indiquem. Esta mudança futura pode ser seguida com identificação de bug Cisco [CSCuo93379](https://www.cisco.com/cisco/web/bugtools/bugsearch.html?bugid=CSCuo93379).

Converta a configuração quando você promove de uma liberação mais cedo do que liberam 5.1.1 para liberar 5.1.1 ou mais atrasado

- MTU padrão:

Se havia uma interface principal sem nenhuma secundário-relação e sem nenhum comando mtu em uma liberação mais cedo do que a liberação 5.1.1:

```
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)
```

E esta relação transporta o dot1q ou os quadros de QinQ, a seguir o MTU deve manualmente ser configurado a "MTU 1522" na liberação 5.1.1 e mais atrasado:

```
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)
```

Esta configuração permite que os quadros de QinQ sejam transportados como nas versões anterior. O valor MTU poderia ser configurado a 1518 se somente o dot1q e não QinQ devem ser transportado.

Se havia umas subinterfaces configuradas para o dot1q ou o QinQ, mas com o "algum" a palavra-chave e nenhuma subinterfaces de QinQ com as 2 etiquetas explícitas foram configuradas em uma liberação mais cedo do que a liberação 5.1.1:

```
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)
```

Esta configuração na liberação 5.1.1 e mais atrasado permitirá somente aos frames do transporte com uma etiqueta assim que o MTU deve igualmente ser aumentado manualmente por 4 bytes se os quadros de QinQ devem ser transportado:

```
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)
```

Se uma secundário-relação de QinQ com as 2 etiquetas explícitas (de que não usa “nenhuma” palavra-chave) é configurada, não há nenhuma necessidade de alterar a configuração MTU quando você promove para liberar 5.1.1 e mais atrasado:

```
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)
```

Se não há nenhuma secundário-relação do transporte L2 mas somente interfaces roteada L3, espera-se que a configuração MTU combinaria em ambos os lados e não haveria uns quadros maiores do que o MTU que é transportado. Não há nenhuma necessidade de atualizar a configuração MTU quando você promove para liberar 5.1.1 e mais atrasado.

- MTU não-padrão na liberação mais cedo do que a liberação 5.1.1:

Similarmente, quando um MTU não-padrão foi configurado em uma liberação mais cedo do que liberam 5.1.1 e nenhuma secundário-relação foi configurada e o dot1q ou os quadros de QinQ têm que ser transportados, a seguir o valor configurado MTU deve ser aumentado por 8 bytes quando você promove para liberar 5.1.1 ou mais atrasado.

Libere mais cedo do que a liberação 5.1.1:

```
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)
```

O MTU deve manualmente ser aumentado por 8 bytes quando você promove para liberar 5.1.1 e mais atrasado:

```
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)

O valor configurado MTU deve igualmente ser aumentado por 4 bytes se há uma secundário-relação do dot1q e nenhuma secundário-relação de QinQ ou uma secundário-relação de QinQ com a qualquer palavra-chave para a etiqueta second-dot1q.

Libere mais cedo do que a liberação 5.1.1:

```
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)

Liberação 5.1.1 e mais atrasado:

```
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)

Se uma secundário-relação de QinQ com as 2 etiquetas explícitas (de que não usa “nenhuma” palavra-chave) é configurada, não há nenhuma necessidade de alterar a configuração MTU quando você promove para liberar 5.1.1 e mais atrasado.

```
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)

Se não há nenhuma secundário-relação do transporte L2, mas somente as interfaces roteada L3, espera-se que a configuração MTU combinaria em ambos os lados e não haveria uns quadros maiores do que o MTU que é transportado. Não há nenhuma necessidade de atualizar a configuração MTU quando você promove para liberar 5.1.1 e mais atrasado.