

Comportamento de MTU no Cisco IOS XR e nos Cisco IOS Routers

Contents

[Introduction](#)

[Informações de Apoio](#)

[Configurar](#)

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

[Interfaces L3 roteadas](#)

[MTU padrão](#)

[MTU não padrão](#)

[Subinterfaces L3 roteadas](#)

[Interface L2VPN L2](#)

[EVC \(ASR9000\)](#)

[Não EVC \(XR 12000 e CRS\)](#)

[Configuração de MTU e MRU do Driver de Interface Ethernet Automática](#)

[Converter a configuração ao atualizar de uma versão anterior à versão 5.1.1 para a versão 5.1.1 ou posterior](#)

Introduction

Este documento descreve os comportamentos da unidade máxima de transmissão (MTU) nos roteadores Cisco IOS[®] XR e compara esses comportamentos com os roteadores Cisco IOS. Ele também discute MTUs em interfaces roteadas da Camada 3 (L3) e interfaces da Camada 2 VPN (L2VPN) L2 que usam os modelos Ethernet Virtual Connection (EVC) e non-EVC. Este documento também descreve alterações importantes em como a MTU do driver da interface Ethernet e a unidade máxima de recepção (MRU) são configuradas automaticamente na versão 5.1.1 e posterior.

Informações de Apoio

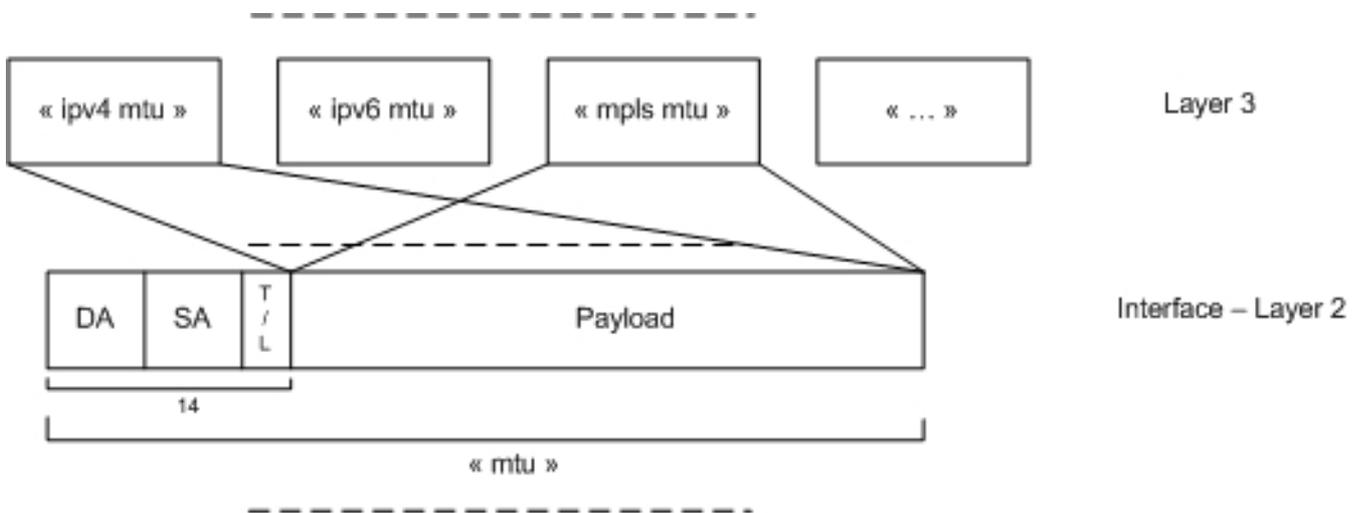
Em redes de computadores, a MTU de um protocolo de comunicação de uma camada define o tamanho, em bytes, da maior unidade de dados de protocolo que a camada tem permissão para transmitir em uma interface. Um parâmetro de MTU é associado a cada interface, camada e protocolo.

As características de MTU no software Cisco IOS XR são:

- Os comandos MTU **config** e **show**, em L2 e L3, incluem o tamanho do cabeçalho de sua camada correspondente. Por exemplo, o comando **mtu** que configura o MTU L2 inclui 14

bytes para uma interface Ethernet (sem dot1q), ou 4 bytes para Protocolo Ponto-a-Ponto (PPP) ou Controle de Enlace de Dados de Alto Nível (HDLC). O comando **ipv4 mtu** inclui 20 bytes do cabeçalho IPv4.

- A MTU de uma camada superior deve caber dentro do payload da camada inferior. Por exemplo, se o MTU da interface de uma interface Ethernet que não seja dot1q for o padrão de 1514 bytes, os protocolos de camada superior, como Multiprotocol Label Switching (MPLS), poderão ter um MTU máximo de 1500 bytes nessa interface. Isso significa que você pode ajustar apenas um quadro MPLS de 1500 bytes (incluindo rótulos) dentro do quadro Ethernet. Você não pode configurar um MPLS MTU de 1508 bytes nessa interface se quiser permitir duas tags MPLS sobre um pacote IPv4 de 1500 bytes. Para transmitir um quadro MPLS de 1508 bytes em uma interface Ethernet, o MTU da interface deve ser aumentado para 1522, ou valor mais alto, para garantir que o payload da interface L2 seja grande o suficiente para transportar o quadro MPLS.



- No software Cisco IOS clássico (não no software Cisco IOS XR), o comando **interface mtu** configura o tamanho de payload da L2, mas não inclui o cabeçalho da L2. Isso é diferente do software Cisco IOS XR, que inclui a sobrecarga de L2 e L3 no comando **interface mtu**. Os comandos MTU de L3, como no caso do comando **ipv4 mtu**, configuram o tamanho máximo do pacote desse protocolo que inclui o cabeçalho de L3. Isso é semelhante ao caso do software Cisco IOS XR.
- O MTU da interface padrão no software Cisco IOS XR deve permitir o transporte de um pacote L3 de 1.500 bytes. Portanto, o MTU padrão é 1514 bytes para uma interface Ethernet principal e 1504 bytes para uma interface serial.

O restante deste documento ilustra as características de MTU, compara o comportamento dos softwares Cisco IOS e Cisco IOS XR e dá exemplos para estes tipos de interfaces:

- Interfaces L3 roteadas
- Subinterfaces L3 roteadas
- Interfaces L2 L2VPN

Configurar

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

Observação: a [Output Interpreter Tool](#) (somente clientes registrados) suporta determinados comandos [show](#). Use a ferramenta Output Interpreter para visualizar uma análise do resultado gerado pelo comando show..

Comparação do software Cisco IOS e Cisco IOS XR

Esta seção compara o comportamento dos softwares Cisco IOS e Cisco IOS XR com referência às características de MTU.

No software Cisco IOS, o comando **mtu** e os comandos **show** correspondentes não incluem o cabeçalho L2. Use o comando **mtu** para configurar o payload L2 para o tamanho máximo dos pacotes L3, incluindo o cabeçalho L3.

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

Se um roteador Cisco IOS estiver configurado com mtu x e conectado a um roteador Cisco IOS XR, a interface correspondente no roteador Cisco IOS XR deverá ser configurada com mtu $x+14$ para interfaces Ethernet ou mtu $x+4$ para interfaces seriais.

Os softwares Cisco IOS e Cisco IOS XR têm o mesmo significado para os comandos **ipv4 mtu**, **ipv6 mtu** e **mpls mtu**; eles devem ser configurados com os mesmos valores.

Como resultado, esta é a configuração no software Cisco IOS 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
```

Interfaces L3 roteadas

Os valores de MTU devem ser os mesmos em todos os dispositivos conectados a uma rede L2. Caso contrário, estes sintomas podem ser relatados:

- As adjacências de Sistema Intermediário para Sistema Intermediário (IS-IS) não aparecem. Por padrão, o IS-IS usa o preenchimento de saudação; portanto, as saudações podem ser caracterizadas como gigantes e podem ser descartadas quando um roteador tem um valor de MTU que é inferior aos valores nos outros roteadores.
- As adjacências Open Shortest Path First (OSPF) ficam presas no estado Exstart ou Exchange, porque os grandes pacotes de descritor de banco de dados (DBD) podem ser

caracterizados como gigantes e podem ser descartados. Quando os pacotes são recebidos em um roteador com um valor de MTU mais baixo, os bancos de dados não são sincronizados.

- O tráfego de dados é caracterizado como gigantes e é descartado quando é recebido em um dispositivo com um valor de MTU inferior ao do dispositivo transmissor.
- Há um throughput baixo quando pacotes grandes são descartados. No caso da descoberta de MTU de caminho, a sessão TCP pode se recuperar quando pacotes grandes são descartados, mas isso afeta o throughput.

MTU padrão

Esta seção analisa o MTU padrão de uma interface roteada quando o comando `mtu` não está 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, o MTU da interface L2 padrão é de 1514 bytes e inclui 14 bytes de cabeçalho Ethernet. Os 14 bytes são contabilizados por 6 bytes de endereço MAC de destino, 6 bytes de endereço MAC de origem e 2 bytes de tipo ou comprimento. Isso não inclui o preâmbulo, o delimitador de quadro, 4 bytes de sequência de verificação de quadro (FCS) e intervalo entre quadros. Para um quadro PPP ou HDLC, 4 bytes do cabeçalho L2 são considerados; portanto, o MTU da interface padrão é de 1504 bytes.

Os protocolos filho L3 herdam sua MTU do payload da MTU pai. Quando você subtrai 14 bytes de um cabeçalho L2 de um MTU L2 de 1514 bytes, você tem um payload L2 de 1500 bytes. Isso se torna o MTU para protocolos L3. IPv4, IPv6, MPLS e Serviço de Rede Sem Conexão (CLNS) herdam esse MTU de 1.500 bytes. Como resultado, uma interface Ethernet Cisco IOS XR, por padrão, pode transportar um pacote L3 de 1.500 bytes que é o mesmo que o default em uma interface Ethernet Cisco IOS.

MTU não padrão

Esta seção mostra como configurar um **mpls mtu** de 1508 para enviar um pacote IPv4 de 1500 bytes com duas tags MPLS de 4 bytes cada, 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 **mpls mtu 1508** seja confirmado, ele não é aplicado, porque o MPLS ainda tem um MTU de 1500 bytes no comando **show**. Isso ocorre porque os protocolos filhos de L3 não podem ter uma MTU maior que o payload da interface L2 pai.

Para permitir dois rótulos em cima de um pacote IP de 1500 bytes, você deve:

- Configure um MTU de interface L2 de 1522 bytes, de modo que todos os protocolos filho (incluindo o MPLS) herdem um MTU de 1508 bytes ($1522 - 14 = 1508$).
- Reduza o MTU dos protocolos L3 para 1500 bytes, de modo que somente o MPLS possa 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 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#
```

Essa configuração permite enviar pacotes IPv4 e IPv6 de 1500 bytes e pacotes MPLS de 1508 bytes (um pacote de 1500 bytes com duas marcas na parte superior).

Subinterfaces L3 roteadas

Essas características se aplicam a subinterfaces L3 roteadas.

Uma MTU de subinterface roteada herda a MTU de sua interface principal pai; adicione 4 bytes para cada marca de VLAN configurada na subinterface. Portanto, há 4 bytes para uma subinterface dot1q e 8 bytes para uma subinterface de encapsulamento IEEE 802.1Q (QinQ).

Como resultado, pacotes L3 do mesmo tamanho podem ser encaminhados tanto na interface principal quanto na subinterface.

O comando **mtu** pode ser configurado na sub-interface, mas é aplicado somente se for menor ou igual ao MTU que é herdado da interface principal.

Este é um exemplo em que o MTU da interface principal é de 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 sub-interface é 2004; adicione 4 bytes ao MTU da interface principal porque há uma marca dot1q configurada na sub-interface.

No entanto, o MTU dos pacotes IPv4 e IPv6 ainda é o mesmo da interface principal (1986). Isso ocorre porque o MTU dos protocolos L3 agora é computado como: $2004 - 14 - 4 = 1986$.

O comando **mtu** pode ser configurado na sub-interface, mas o MTU configurado será aplicado somente se for menor ou igual ao MTU que é herdado da interface principal (4 bytes maior que o MTU da interface principal).

Quando o MTU da subinterface for maior que o MTU herdado, ele não será 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** para reduzir o valor de MTU herdado da interface principal.

Da mesma forma, você também pode usar os comandos MTU dos protocolos L3 (IPv4, IPv6, MPLS) para reduzir o valor da MTU L3 herdada da carga útil da subinterface L2. A MTU do protocolo L3 não tem efeito quando é configurada para um valor que não se encaixa no payload da MTU L2.

Interface L2VPN L2

O MTU para um L2VPN é importante porque o Label Distribution Protocol (LDP) não ativa um pseudofio (PW) quando os MTUs nos circuitos de conexão em cada lado de um PW não são os mesmos.

Aqui está um comando **show** que ilustra que um PW L2VPN permanece inativo quando há uma incompatibilidade de 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, observe que as bordas do provedor de MPLS L2VPN (PEs) em cada lado devem sinalizar o mesmo valor de MTU para ativar o PW.

A MTU sinalizada pelo LDP de MPLS não inclui a sobrecarga de L2. Isso é diferente dos comandos XR interface **config** e **show** que incluem a sobrecarga de L2. O MTU na subinterface é de 2018 bytes (como herdado da interface principal de 2014 bytes), mas o LDP sinalizou um MTU de 2000 bytes. Como resultado, ele subtrai 18 bytes (14 bytes de cabeçalho Ethernet + 4 bytes de 1 tag dot1q) do cabeçalho L2.

É importante entender como cada dispositivo calcula os valores de MTU dos circuitos de conexão para corrigir incompatibilidades de MTU. Isso depende de parâmetros como fornecedor, plataforma, versão de software e configuração.

EVC (ASR9000)

O Cisco ASR 9000 Series Aggregation Services Router usa o modelo de infraestrutura EVC, que permite a correspondência flexível de VLAN em interfaces L2VPN L2 e subinterfaces.

As interfaces L2VPN L2 do EVC têm estas características:

- Eles permitem a configuração de uma ou mais tags com o comando **encapsulation**.
- Por padrão e apenas com o comando **encapsulation**, as tags são preservadas e transportadas por PWs. Como resultado, não é necessário remover tags por padrão, como é necessário em plataformas não EVC.
- Use o comando **rewrite** quando decidir soltar as tags de entrada ou empurrar algumas tags adicionais sobre o quadro de entrada.

Para calcular a MTU da subinterface, pegue a MTU da interface principal (o padrão ou aquele configurado manualmente na interface principal) e adicione 4 bytes para cada marca de VLAN configurada com o comando **encapsulation**. Consulte [Comandos Específicos de Encapsulamento EFP](#).

Quando há um comando **mtu** na sub-interface, ele tem efeito somente se for inferior ao MTU calculado. O comando **rewrite** não influencia a MTU da sub-interface.

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 tem 2014 bytes; adicione 8 bytes porque há duas marcas configuradas na subinterface.

Se você configurar **mtu 2026** bytes sob a subinterface, ela não será aplicada porque é maior que a MTU da subinterface herdada da interface principal (2022). Como resultado, você pode configurar apenas uma MTU da subinterface inferior a 2022 bytes.

Com base nesse MTU de sub-interface, calcule o MTU do payload LDP de MPLS que é sinalizado para o vizinho e certifique-se de que seja idêntico ao computado pelo PE de L2VPN remoto. É aqui que o comando **rewrite** entra em ação.

Para calcular a MTU da carga útil do LDP do MPLS, pegue a MTU da sub-interface e:

1. Subtraia 14 bytes do cabeçalho Ethernet.
2. Subtraia 4 bytes para cada marca exibida no comando **rewrite** configurado na sub-interface.
3. Adicione 4 bytes para cada marca enviada no comando **rewrite** configurado na sub-interface.

Este é o mesmo exemplo com a configuração QinQ em gig 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)
```

Este é o cálculo para o MTU da carga útil do LDP do MPLS:

1. Valor de MTU da subinterface: 2022 bytes
2. Subtraia 14 bytes do cabeçalho Ethernet: $2022 - 14 = 2008$ bytes
3. Subtraia 4 bytes para cada marca exibida no **reescrever** comando: $2008 - 4 * 2 = 2000$

Assegure-se de que o lado remoto anuncie um payload LDP MPLS de 2000 bytes. Caso contrário, ajuste o tamanho da MTU do circuito de conexão local ou remoto (AC) para que corresponda.

```
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
```

Comandos de Encapsulamento Específicos do Ponto de Fluxo Ethernet (EFP - Ethernet Flow Point)

Esses encapsulamentos contam como tags zero correspondentes, portanto não aumentam o MTU da sub-interface:

- **encapsulamento não marcado**
- **encapsulation default**

Esses modificadores de encapsulamento não afetam o número de tags necessárias para calcular a MTU da subinterface:

- **nativo**
- **payload-ethertype**

- exato
- cos
- ingress source-mac ou ingress destination-mac

encapsulation [dot1q|dot1ad] priority-tagged conta como correspondendo a uma única marca.

A palavra-chave 'any' usada como a correspondência de marca mais interna não aumenta a MTU da subinterface.

- **encapsulation dot1q any** não aumenta a MTU da sub-interface.
- o **encapsulamento dot1ad 10 dot1q any any** é considerado como uma tag; ele aumenta o MTU da sub-interface em 4 bytes.
- o **encapsulamento dot1ad any dot1q 7** é considerado como duas tags; ele aumenta o MTU da sub-interface em 8 bytes.

Os intervalos de VLAN-IDs incrementam a MTU da subinterface:

- **encapsulation dot1q 10-100** é considerado como uma tag; ele aumenta o MTU da sub-interface em 4 bytes.

A sobrecarga de MTU de encapsulamento de um EFP que é uma correspondência disjuntiva é tratada como a MTU de seu elemento mais alto.

- **encapsulation dot1q 10-100, untagged** é contabilizado como uma tag porque o intervalo 10 - 100 é o elemento mais alto.

Não EVC (XR 12000 e CRS)

Roteadores como o Cisco XR 12000 Series Router e o Carrier Routing System (CRS) usam a configuração tradicional para a correspondência de VLAN em subinterfaces. Essas características se aplicam às interfaces L2VPN L2 em CRS e em roteadores XR 12000 que não seguem o modelo EVC:

- Em plataformas não EVC, as tags dot1q ou dot1ad de entrada são automaticamente removidas quando são recebidas em uma subinterface de transporte L2.
- Ao calcular o tamanho do payload para o LDP do MPLS sinalizar, subtraia o tamanho das tags da MTU da subinterface, conforme visto no comando **show interface**.
- Isso é semelhante ao caso de uma subinterface roteada.
- A sub-interface herda seu MTU da interface principal; adicione os 4 bytes de cada tag ao MTU da interface principal para computar o MTU da sub-interface. Por exemplo, se uma subinterface QinQ tiver 2 marcas dot1q, a subinterface, por padrão, terá uma MTU que é 8 bytes maior que a MTU da interface principal.
- Você também pode usar o comando **mtu** na sub-interface, mas ele é usado apenas para reduzir o MTU da sub-interface, que é herdado do MTU da interface principal.

Aqui estão vários exemplos que ilustram essas características.

Este exemplo mostra como uma subinterface não 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 não EVC usam os comandos **dot1q vlan** ou **dot1ad vlan** em vez dos comandos **encapsulation** e **rewrite** das plataformas EVC (ASR9000).

Se você não configurar uma MTU explicitamente na interface principal ou na sub-interface, um pacote L3 de 1.500 bytes poderá ser recebido por padrão:

```
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 MTU da subinterface é calculada a partir da MTU da interface principal (1514); adicione 4 bytes para cada tag dot1q. Como há uma tag configurada na subinterface com o comando **dot1q vlan 201**, adicione 4 bytes a 1514 para um MTU de 1518 bytes.

O MTU de payload correspondente no LDP do MPLS é de 1500 bytes, já que os 14 bytes do cabeçalho Ethernet não são contados e a tag one dot1q é exibida automaticamente pela plataforma não EVC quando passa pelo PW:

```
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 aumentar o MTU da interface principal para 2014 bytes, o MTU da subinterface será aumentado de acordo:

```
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, para calcular a MTU LDP do MPLS, subtraia 14 bytes do cabeçalho Ethernet e adicione 4 bytes para cada tag configurada na subinterface.

Configuração de MTU e MRU do Driver de Interface Ethernet Automática

Nas interfaces Ethernet, o driver da interface é configurado com um MTU e um MRU que se baseiam na configuração de MTU da interface.

O MTU e o MRU configurados no driver de interface Ethernet podem ser vistos com o comando **show controller <interface> all**.

Em versões anteriores ao Cisco IOS XR Release 5.1.1, o MTU e o MRU no driver da interface Ethernet foram configurados automaticamente com base na configuração do Cisco IOS XR MTU na interface.

A MTU/MRU configurada no driver Ethernet era simplesmente baseada na MTU configurada + 12 bytes para a adição de 2 marcações Ethernet e o campo CRC. Os 12 bytes foram adicionados ao driver Ethernet MTU/MRU, independentemente de haver alguma marca de VLAN configurada nas subinterfaces.

Um exemplo com todas as versões do Cisco IOS XR anteriores ao Cisco IOS XR Release 5.1.1 e um MTU padrão de 1514 em uma interface 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 Release 5.1.1 e posterior, o MTU e o MRU usados no driver da interface Ethernet foram alterados e agora se baseiam no número de tags de VLAN configuradas em qualquer uma das subinterfaces.

Se nenhuma marca de VLAN for configurada em nenhuma subinterface, o driver MTU/MRU será igual ao MTU configurado na interface + 4 bytes de CRC, por exemplo, 1514 + 4 = 1518 bytes.

Se uma VLAN for configurada em qualquer subinterface, o driver MTU/MRU será igual ao MTU configurado + 8 bytes (1 tag + CRC), por exemplo 1514 + 8 = 1522 bytes.

Se duas marcas de VLAN forem configuradas em qualquer subinterface, o driver MTU/MRU será igual ao MTU configurado + 12 bytes (2 marcas + CRC), por exemplo 1514 + 12 = 1526 bytes

Se QinQ com a palavra-chave **any** estiver configurado para a tag second-dot1q, o driver MTU/MRU será igual ao MTU configurado + 8 bytes (1 tag + CRC), por exemplo 1514 + 8 = 1522 bytes.

Estes exemplos exibem o comportamento no Cisco IOS XR Release 5.1.1 e posterior 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 12transport
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, essa alteração de comportamento na Versão 5.1.1 e posterior não deve exigir nenhuma alteração na configuração da MTU na interface.

Essa alteração de comportamento pode causar problemas no caso de uma subinterface configurada com uma única marca de VLAN, mas recebe pacotes com duas marcas de VLAN. Nessa situação, os pacotes recebidos podem exceder o MRU no driver da interface Ethernet. Para eliminar essa condição, o MTU da interface pode ser aumentado em 4 bytes ou a subinterface configurada com duas marcas de VLAN.

A configuração automática de MTU e MRU do driver da interface Ethernet no comportamento da versão 5.1.1 é a mesma para roteadores CRS e ASR 9000. Mas um roteador CRS que executa a versão 5.1.1 não inclui o CRC de 4 bytes no valor de MTU e MRU exibido na saída de **show controller**. O comportamento de como ele é relatado não é o mesmo entre o CRS e o 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 forma como a MTU e a MRU são exibidas na saída do comando show controller no ASR 9000 será alterada no futuro, de modo que os 4 bytes de CRC não serão incluídos no valor de MTU/MRU exibido. Essa alteração futura pode ser rastreada com o bug da Cisco ID [CSCuo93379](https://tools.cisco.com/bugcenter/bug/?bugID=CSCuo93379).

Converter a configuração ao atualizar de uma versão anterior à versão 5.1.1 para a versão 5.1.1 ou posterior

- MTU padrão:

Se houver uma interface principal sem nenhuma subinterface e sem nenhum comando `mtu` em uma versão anterior à Versão 5.1.1:

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

E essa interface transporta quadros dot1q ou QinQ, então o MTU deve ser manualmente configurado para "mtu 1522" na Versão 5.1.1 e posterior:

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

Essa configuração permite que os quadros QinQ sejam transportados como nas versões anteriores. O valor de MTU poderia ser configurado como 1518 se apenas dot1q e não QinQ fossem transportados.

Se houver subinterfaces configuradas para dot1q ou QinQ, mas com a palavra-chave "any" e nenhuma subinterface QinQ com 2 marcas explícitas foi configurada em uma versão anterior à 5.1.1:

```
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
!
```

Essa configuração na versão 5.1.1 e posterior permitirá apenas o transporte de quadros com uma tag, de modo que o MTU também deve ser aumentado manualmente em 4 bytes se os quadros QinQ forem transportados:

```
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
!
```

Se uma sub-interface QinQ com 2 tags explícitas (que não usam a palavra-chave "any") estiver configurada, não haverá necessidade de modificar a configuração de MTU ao atualizar para a Versão 5.1.1 e posterior:

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

Se não houver subinterface de transporte L2, mas apenas interfaces roteadas L3, espera-se que a configuração de MTU corresponda em ambos os lados e não haverá quadros maiores do que a MTU transportada. Não há necessidade de atualizar a configuração de MTU ao atualizar para a Versão 5.1.1 e posterior.

- MTU não padrão na versão anterior à 5.1.1:

Da mesma forma, quando uma MTU não padrão foi configurada em uma versão anterior à 5.1.1 e nenhuma subinterface foi configurada e os quadros dot1q ou QinQ precisam ser transportados, o valor de MTU configurado deve ser aumentado em 8 bytes quando você atualizar para a Versão 5.1.1 ou posterior.

Versão anterior à Versão 5.1.1:

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

O MTU deve ser aumentado manualmente em 8 bytes quando você atualizar para a Versão 5.1.1 e posterior:

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

O valor de MTU configurado também deve ser aumentado em 4 bytes se houver uma subinterface dot1q e nenhuma subinterface QinQ ou uma subinterface QinQ com a palavra-chave any para a tag second-dot1q.

Versão anterior à Versão 5.1.1:

```
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  
!
```

Versão 5.1.1 ou posterior:

```
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  
!
```

Se uma subinterface QinQ com 2 tags explícitas (que não usam a palavra-chave "any") estiver configurada, não haverá necessidade de modificar a configuração de MTU quando você atualizar para a Versão 5.1.1 e posterior.

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

Se não houver nenhuma subinterface de transporte L2, mas apenas interfaces roteadas L3, espera-se que a configuração de MTU corresponda em ambos os lados e que não haja quadros maiores que a MTU transportada. Não há necessidade de atualizar a configuração de MTU ao atualizar para a Versão 5.1.1 e posterior.

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.