

Comportamiento MTU en el Cisco IOS XR y el Routers del Cisco IOS

Contenido

[Introducción](#)

[Antecedentes](#)

[Configurar](#)

[Comparación del Cisco IOS y del Software Cisco IOS XR](#)

[Interfaces ruteadas L3](#)

[MTU predeterminado](#)

[No valor por defecto MTU](#)

[Subinterfaces ruteados L3](#)

[Interfaz L2VPN L2](#)

[EVC \(ASR9000\)](#)

[No EVC \(XR 12000 y CRS\)](#)

[Driver automático MTU de la interfaz de Ethernet y configuración MRU](#)

[Convierta la configuración cuando usted actualiza de una versión anterior que 5.1.1 para liberar 5.1.1 o más adelante](#)

Introducción

Este documento describe los comportamientos de la Unidad máxima de transmisión (MTU) (MTU) en el Routers del [®] XR del Cisco IOS y compara esos comportamientos al Routers del Cisco IOS. Él también discute los MTU en las interfaces ruteadas de la capa 3 (L3) y las interfaces L2 de la capa 2 VPN (L2VPN) que utilizan los modelos conexión virtual de los Ethernetes (EVC) y NON-EVC. Este documento también describe los cambios importantes a cómo el driver MTU de la interfaz de Ethernet y el Maximum Receive Unit (MRU) se configuran automáticamente en la versión 5.1.1 y posterior.

Antecedentes

En el establecimiento de una red del ordenador, el MTU de un protocolo de comunicaciones de una capa define el tamaño, en los bytes, de la unidad de datos de protocolo más grande que la capa se permite transmitir sobre una interfaz. Un parámetro MTU se asocia a cada interfaz, capa, y protocolo.

Las características MTU en el Software Cisco IOS XR son:

- **Los config** y los **comandos show** MTU, en el L2 y el L3, incluyen el tamaño de encabezado de su capa correspondiente. Por ejemplo, el **comando mtu** que configura el L2 MTU incluye 14

bytes para una interfaz de Ethernet (sin el dot1q), o 4 bytes para el Point-to-Point Protocol (PPP) o el High-Level Data Link Control (HDLC). **El comando mtu ipv4** incluye 20 bytes de la encabezado del IPv4.

- El MTU de una capa superior debe caber dentro del payload de la capa inferior. Por ejemplo, si el MTU de interfaz de una interfaz de Ethernet non-dot1q es el valor por defecto de 1514 bytes, después los protocolos de capa más altas tales como Multiprotocol Label Switching (MPLS) pueden tener un MTU máximo de 1500 bytes en esa interfaz. Esto significa que usted puede caber solamente una trama de 1500 bytes MPLS (escrituras de la etiqueta incluyendo) dentro de la trama Ethernet. Usted no puede configurar 1508 un byte MPLS MTU en esa interfaz si usted quiere permitir dos etiquetas MPLS encima de un byte 1500 paquete IPV4. Para transmitir una trama de 1508 bytes MPLS en una interfaz de Ethernet, el MTU de interfaz se debe aumentar a 1522, o a un valor más alto, para asegurarse de que el payload de la interfaz L2 es bastante grande llevar la trama MPLS.
- En el Cisco IOS Software clásico (no el Software Cisco IOS XR), el **comando mtu de la interfaz** configura el Tamaño de carga útil L2, pero no incluye la encabezado L2. El es diferente del Software Cisco IOS XR que incluye los gastos indirectos L2 y L3 en el **comando mtu de la interfaz**. Los comandos mtu L3, como en el caso del **comando mtu ipv4**, configuran el tamaño máximo de paquete de ese protocolo que incluya la encabezado L3. Esto es similar al caso del Software Cisco IOS XR.
- La interfaz predeterminada MTU en el Software Cisco IOS XR debe permitir el transporte de un paquete de 1500 bytes L3. Por lo tanto, el MTU predeterminado es 1514 bytes para una interfaz de Ethernet principal y 1504 bytes para una interfaz serial.

El recordatorio de este documento ilustra las características MTU, compara el comportamiento del Cisco IOS y del Software Cisco IOS XR, y da los ejemplos para estos tipos de interfaces:

- Interfaces ruteadas L3
- Subinterfaces ruteados L3
- Interfaces L2VPN L2

Configurar

Nota: Use la [Command Lookup Tool](#) ([clientes registrados solamente](#)) para obtener más información sobre los comandos usados en esta sección.

Nota: [La herramienta del Output Interpreter](#) ([clientes registrados solamente](#)) apoya los ciertos comandos show. Utilice la herramienta del Output Interpreter para ver una análisis de la salida del comando show.

Comparación del Cisco IOS y del Software Cisco IOS XR

Esta sección compara el comportamiento del Cisco IOS y del Software Cisco IOS XR referente a las características MTU.

En Cisco IOS Software, el **comando mtu** y los **comandos show** correspondientes no incluyen la encabezado L2. Utilice el **comando mtu** para configurar el payload L2 al tamaño máximo para los paquetes L3, incluyendo la encabezado L3.

Esto es diferente del Software Cisco IOS XR, donde el **comando mtu** incluye la encabezado L2 (14 bytes para Ethernet o 4 bytes para el PPP/HDLC).

Si configuran con **MTU x** y están conectado a un router del Cisco IOS con un router del Cisco IOS XR, después la interfaz correspondiente en el router del Cisco IOS XR se debe configurar con **MTU x+14** para las interfaces de Ethernet, o **MTU x+4** para las interfaces seriales.

El Cisco IOS y el Software Cisco IOS XR tienen el mismo significado para el **MTU ipv4**, el **MTU del IPv6** y los **comandos mpls mtu**; él debe ser configurado con los mismos valores.

Como consecuencia, ésta es la configuración en Cisco IOS Software en una interfaz de Ethernet:

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

La configuración correspondiente en el vecino del Software Cisco IOS XR es:

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

Interfaces ruteadas L3

Los valores MTU deben ser lo mismo en todos los dispositivos conectados con una red L2. Si no, estos síntomas pudieron ser señalados:

- Las adyacencias del Intermediate System-to-Intermediate System (IS-IS) no suben. Por abandono, el IS-IS utiliza el relleno de saludo; por lo tanto, el hellos se pudo caracterizar como gigantes y pudo ser caído cuando un router tiene un valor MTU que sea más bajo que los valores en el otro Routers.
- Las adyacencias del Open Shortest Path First (OSPF) consiguen pegadas en el exstart o el estado de intercambio, porque los paquetes grandes del Database Descriptor (DBD) se pudieron caracterizar como gigantes y pudieron ser caídos. Cuando los paquetes se reciben en un router con un valor del MTU inferior, las bases de datos no se sincronizan.
- El tráfico de datos se caracteriza como gigantes y se cae cuando se recibe en un dispositivo con un valor MTU que sea más bajo que el que está en el dispositivo que transmite.
- Hay bajo rendimiento cuando los paquetes grandes consiguen caídos. En caso de la detección de MTU de trayecto, la sesión TCP puede recuperarse cuando se caen los paquetes grandes, pero éste afecta a la producción.

MTU predeterminado

Esta sección analiza el MTU predeterminado de una interfaz ruteada cuando no configuran al **comando mtu**:

```
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: 0x00000000010059f (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#
```

En este ejemplo, el MTU de interfaz del valor por defecto L2 es 1514 bytes e incluye 14 bytes del encabezado Ethernet. Los 14 bytes son considerados por 6 bytes de la dirección MAC del destino, 6 bytes del MAC Address de origen, y 2 bytes del tipo o de la longitud. Esto no incluye el preámbulo, el delimitador de la trama, 4 bytes de Secuencia de verificación de tramas (FCS), y el intervalo de la inter-trama. Para una trama PPP o del HDLC, 4 bytes de la encabezado L2 se explican; la interfaz predeterminada MTU es tan 1504 bytes.

Los protocolos del niño L3 heredan su MTU del payload del padre MTU. Cuando usted resta 14 bytes de una encabezado L2 de un L2 MTU de 1514 bytes, usted tiene un payload L2 de 1500 bytes. Éste se convierte en el MTU para los protocolos L3. El IPv4, el IPv6, el MPLS, y el servicio de red sin conexión (CLNS) heredan this1500 el byte MTU. Como consecuencia, una interfaz de Ethernet del Cisco IOS XR, por abandono, puede transportar un paquete de 1500 bytes L3 que sea lo mismo que el default en una interfaz de Ethernet del Cisco IOS.

No valor por defecto MTU

Esta sección muestra cómo configurar un **MTU de los mpls** de 1508 para enviar paquete IPV4 de 1500 bytes con dos etiquetas MPLS de 4 bytes cada uno, encima del paquete:

```
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#

Aunque el comando **MTU 1508 de los mpls** esté confiado, no es aplicado, porque el MPLS todavía tiene un MTU de 1500 bytes en el **comando show**. Esto es porque los protocolos del niño L3 no pueden tener un MTU más grande que el payload de su interfaz del padre L2.

Para permitir dos escrituras de la etiqueta encima de 1500 byte ip paquete, usted debe:

- Configure un MTU de interfaz L2 de 1522 bytes, de modo que todos los protocolos del niño (MPLS incluyendo) hereden un MTU de 1508 bytes (1522 - 14 = 1508).
- Reduzca el MTU de los protocolos L3 a 1500 bytes, para solamente permitir el MPLS que exceda 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#
```

Esta configuración le deja enviar los paquetes del IPv4 y del IPv6 de 1500 bytes y los paquetes MPLS de 1508 bytes (un paquete de bytes 1500 con dos etiquetas en el top).

Subinterfaces ruteados L3

Estas características se aplican a las subinterfaces L3.

Una sub-interfaz ruteada MTU hereda el MTU de su interfaz principal del padre; agregue 4 bytes para cada etiqueta del VLA N configurada en la sub-interfaz. Así, hay 4 bytes para una sub-interfaz del dot1q y 8 bytes para un IEEE 802.1Q que hace un túnel a la sub-interfaz (de QinQ).

Como consecuencia, los paquetes L3 del mismo tamaño se pueden remitir en la interfaz principal y la sub-interfaz.

El comando mtu puede ser configurado bajo sub-interfaz, pero se aplica solamente si es más baja o igual al MTU que se hereda de la interfaz principal.

Esto es un ejemplo donde está 2000 bytes el MTU de la interfaz principal:

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

```
mtu 2000
!
```

```
RP/0/RP0/CPU0:router#sh run int gig 0/1/0/3.100
```

```
interface GigabitEthernet0/1/0/3.100
```

```
ipv4 address 10.0.2.1 255.255.255.0
```

```
ipv6 address 2001:db9:0:1::1/64
```

```
dot1q vlan 100
```

```
!
```

```
RP/0/RP0/CPU0:router#sh int gig 0/1/0/3.100 | i MTU
```

```
MTU 2004 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
```

```
RP/0/RP0/CPU0:router#show im database interface gigabitEthernet 0/1/0/3.100
```

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

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

```
Interface GigabitEthernet0/1/0/3.100, ifh 0x01180260 (up, 2004)
```

```
Interface flags: 0x0000000000000597 (IFINDEX|SUP_NAMED_SUB
```

```
|BROADCAST|CONFIG|VIS|DATA|CONTROL)
```

```
Encapsulation: dot1q
```

```
Interface type: IFT_VLAN_SUBIF
```

```
Control parent: GigabitEthernet0/1/0/3
```

```
Data parent: GigabitEthernet0/1/0/3
```

```
Views: GDP|LDP|L3P|OWN
```

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

```
-----
```

```
None vlan_jump (up, 2004)
```

```
None dot1q (up, 2004)
```

```
arp arp (up, 1986)
```

```
ipv4 ipv4 (up, 1986)
```

```
ipv6 ipv6_preswitch (up, 1986)
```

```
ipv6 ipv6 (down, 1986)
```

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

En los **comandos show**, el MTU de la sub-interfaz es 2004; agregue 4 bytes al MTU de la interfaz principal porque hay una etiqueta del dot1q configurada bajo sub-interfaz.

Sin embargo, el MTU de los paquetes del IPv4 y del IPv6 sigue siendo lo mismo que el de la interfaz principal (1986). Esto es porque el MTU de los protocolos L3 ahora se computa como: $2004 - 14 - 4 = 1986$.

El comando mtu puede ser configurado bajo sub-interfaz, pero se aplica el MTU configurado solamente si es más bajo o igual al MTU que se hereda de la interfaz principal (4 bytes más grande que el MTU de la interfaz principal).

Cuando el MTU de la sub-interfaz que es más grande que el MTU heredado, él no es aplicado, como se muestra aquí:

```
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#
```

Así, usted puede utilizar solamente el **comando mtu** para bajar el valor MTU heredado de la interfaz principal.

Semejantemente, usted puede también utilizar los comandos mtu de los protocolos L3 (IPv4, IPv6, MPLS) para bajar el valor del L3 MTU heredado del payload de la sub-interfaz L2. L3 el protocolo MTU no toma el efecto cuando se configura a un valor que no quepa en el payload del L2 MTU.

Interfaz L2VPN L2

El MTU para un L2VPN es importante porque el Protocolo de distribución de etiquetas (LDP) no trae un pseudowire (picovatio) encima de cuando los MTU en los circuitos de la conexión en cada lado de un picovatio no son lo mismo.

Aquí está un **comando show** que ilustra que un L2VPN picovatio permanece abajo cuando hay una discordancia 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#

En este ejemplo, observe que los bordes del proveedor MPLS L2VPN (PE) en cada lado deben señalar el mismo valor MTU para sacar a colación el picovatio.

El MTU señalado por MPLS LDP no incluye los gastos indirectos L2. Esto es diferente de los **config** y de los **comandos show de la** interfaz XR que incluyen los gastos indirectos L2. El MTU en la sub-interfaz es 2018 bytes (según lo heredado de la interfaz principal de 2014 bytes), pero el LDP señaló un MTU de 2000 bytes. Como consecuencia, resta 18 bytes (14 bytes del encabezado Ethernet + 4 bytes de 1 etiqueta del dot1q) de la encabezado L2.

Es importante entender cómo cada dispositivo computa los valores MTU de los circuitos de la conexión para reparar las discordancias MTU. Esto depende de los parámetros tales como vendedor, plataforma, versión de software, y configuración.

EVC (ASR9000)

La agregación de las 9000 Series de Cisco ASR mantiene al router utiliza el modelo de la infraestructura EVC, que permite el VLA N flexible que corresponde con en las interfaces y los subinterfaces L2VPN L2.

Las interfaces EVC L2VPN L2 tienen estas características:

- Permiten la configuración de una o más etiquetas con el **comando encapsulation**.
- Por abandono y con apenas el **comando encapsulation**, las etiquetas se preservan y se transportan sobre PWs. Como consecuencia, usted no necesita eliminar las etiquetas por abandono, mientras que usted necesita hacer en las Plataformas NON-EVC.
- Utilice el comando de la **reescritura** cuando usted decide hacer estallar las etiquetas entrantes o avanzar algunas etiquetas adicionales encima de la trama entrante.

Para computar la sub-interfaz MTU, tomar la interfaz principal MTU (el valor por defecto o el que está configurado manualmente bajo interfaz principal), y agregar 4 bytes para cada etiqueta del VLA N configurada con el **comando encapsulation**. Vea los [comandos encapsulation específicos EFP](#).

Cuando hay **comando mtu** bajo sub-interfaz, toma el efecto solamente si es más baja que el MTU computado. El comando de la **reescritura** no influencia el MTU de la sub-interfaz.

Aquí tiene un ejemplo:

```
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)
```

En este ejemplo, el MTU en la interfaz principal es 2014 bytes; agregue 8 bytes porque hay dos etiquetas configuradas bajo sub-interfaz.

Si usted configura el **MTU 2026** bytes bajo sub-interfaz, no es aplicada porque es más grande que el MTU de la sub-interfaz heredada de la interfaz principal (2022). Como consecuencia, usted puede configurar solamente una sub-interfaz MTU más bajo de 2022 bytes.

De acuerdo con esta sub-interfaz MTU, compute el MTU del payload MPLS LDP que se señala al vecino, y asegúrese que es idéntico al que está computado por el telecontrol L2VPN PE. Aquí es adonde el comando de la **reescritura** entra en el juego.

Para computar el MTU del payload MPLS LDP, tome el MTU de la sub-interfaz, entonces:

1. Reste 14 bytes para el encabezado Ethernet.
2. Reste 4 bytes para cada etiqueta hecha estallar en el comando de la **reescritura** configurado bajo sub-interfaz.
3. Agregue 4 bytes para cada etiqueta empujada hacia adentro el comando de la **reescritura** configurado bajo sub-interfaz.

Éste es el mismo ejemplo con la configuración de QinQ en el carruaje 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)
```

Éste es el cómputo para el MTU del payload MPLS LDP:

1. Valor MTU de la sub-interfaz MTU: 2022 bytes
2. Reste 14 bytes del encabezado Ethernet: $2022 - 14 = 2008$ bytes
3. Reste 4 bytes para cada etiqueta hecha estallar en el comando de la **reescritura**: $2008 - 4 * 2 = 2000$

Asegúrese de que el lado remoto haga publicidad de un payload MPLS LDP de 2000 bytes. Si no, ajuste la talla del MTU local o remota del circuito de la conexión (AC) así que hacen juego.

```
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
```

Los Ethernetes específicos fluyen los comandos encapsulation de la punta (EFP)

Estas encapsulaciones cuentan como etiquetas cero que corresponden con, así que no aumentan la sub-interfaz MTU:

- **encapsulación untagged**
- **valor por defecto de la encapsulación**

Estos modificantes de la encapsulación no afectan al número de etiquetas requeridas para computar la sub-interfaz MTU:

- nativo
- payload-Ethertype
- exacto
- lechuga romana
- fuente-mac del ingreso o destino-mac del ingreso

cuentas **etiquetadas por prioridad de la encapsulación [dot1q|dot1ad]** como corresponder con una sola etiqueta.

La “ninguna” palabra clave usada como la coincidencia de etiqueta íntima no aumenta la sub-interfaz MTU.

- **el dot1q de la encapsulación** no aumenta la sub-interfaz MTU.
- **el dot1q de la encapsulación dot1ad 10** se considera como una etiqueta; aumenta la sub-interfaz MTU en 4 bytes.
- **la encapsulación dot1ad cualquier dot1q 7** se considera como dos etiquetas; aumenta la sub-interfaz MTU en 8 bytes.

Los rangos de los VLAN-ID incrementan la sub-interfaz MTU:

- se considera el **dot1q 10-100 de la encapsulación** como una etiqueta; aumenta la sub-interfaz MTU en 4 bytes.

Los gastos indirectos de la encapsulación MTU de un EFP que sea una coincidencia disyuntiva se tratan como el MTU de su elemento más alto.

- el dot1q 10-100 de la encapsulación, untagged se considera pues una etiqueta porque el rango 10 -100 es el elemento más alto.

No EVC (XR 12000 y CRS)

El Routers como el Router Cisco XR de la serie 12000 y el sistema de ruteo del portador (CRS) utiliza la configuración tradicional para el VLA N que corresponde con en los subinterfaces. Estas características se aplican a las interfaces L2VPN L2 en CRS y en los 12000 Router XR que no siguen el modelo EVC:

- En las Plataformas NON-EVC, el dot1q entrante o las etiquetas dot1ad se elimina automáticamente cuando se reciben en una sub-interfaz del transporte L2.
- Cuando usted está computando el Tamaño de carga útil para MPLS LDP para señalar, reste el tamaño de las etiquetas del MTU de la sub-interfaz, como se ve en el **comando show interface**.
- Esto es similar al caso de una sub-interfaz ruteada.
- La sub-interfaz hereda su MTU de la interfaz principal; agregue los 4 bytes para cada etiqueta al MTU de la interfaz principal para computar el MTU de la sub-interfaz. Por ejemplo, si una sub-interfaz de QinQ tiene 2 etiquetas del dot1q, la sub-interfaz, por abandono, tiene un MTU que sea 8 bytes más grande que el MTU de la interfaz principal.
- Usted puede también utilizar el **comando mtu** bajo sub-interfaz, pero se utiliza para reducir solamente el MTU de la sub-interfaz, que se hereda del MTU de la interfaz principal.

Aquí están varios ejemplos que ilustran estas características.

Este ejemplo muestra cómo se configura una sub-interfaz NON-EVC:

```
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#
```

Las Plataformas NON-EVC utilizan el **dot1q vlan** o los **comandos vlan dot1ad** en vez de la **encapsulación y reescriben los comandos** de las Plataformas EVC (ASR9000).

Si usted no configura un MTU explícitamente en la tubería o la sub-interfaz, después un paquete de 1500 bytes L3 se puede recibir por abandono:

```
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#
```

La sub-interfaz MTU se computa de la interfaz principal MTU (1514); agregue 4 bytes para cada etiqueta del dot1q. Porque hay una etiqueta configurada en la sub-interfaz con el **dot1q 201 vlan** ordenan, agregan 4 bytes a 1514 para un MTU de 1518 bytes.

El payload correspondiente MTU en MPLS LDP es 1500 bytes, puesto que los 14 bytes del encabezado Ethernet no se cuentan y la una etiqueta del dot1q es hecha estallar automáticamente por la plataforma NON-EVC cuando pasa el picovatio:

```
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
```

Si usted aumenta el MTU de la interfaz principal a 2014 bytes, el MTU de la sub-interfaz se aumenta por consiguiente:

```
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
```

Así pues, para computar el MPLS LDP MTU, restar 14 bytes del encabezado Ethernet, y agregar 4 bytes para cada etiqueta configurada bajo sub-interfaz.

Driver automático MTU de la interfaz de Ethernet y configuración MRU

En las interfaces de Ethernet el driver de la interfaz se configura con un MTU y un MRU que se base en la configuración del MTU de interfaz.

El MTU y el MRU configurados en el driver de la interfaz de Ethernet se pueden ver con el comando **all del <interface> del regulador de la demostración.**

En las versiones anterior que la versión 5.1.1 del Cisco IOS XR, el MTU y el MRU en el driver de la interfaz de Ethernet fueron configurados automáticamente sobre la base de la configuración del Cisco IOS XR MTU en la interfaz.

El MTU/MRU configurado en el driver Ethernet fue basado simplemente en el MTU+ configurado 12 bytes para la adición de 2 etiquetas de los Ethernets y del campo de CRC. Los 12 bytes fueron agregados al driver Ethernet MTU/MRU sin importar si había algunas etiquetas del VLA N configuradas en los subinterfaces.

Un ejemplo con todas las versiones del Cisco IOS XR anterior que la versión 5.1.1 del Cisco IOS

XR y un MTU predeterminado de 1514 en una interfaz ASR 9000 se muestra aquí:

```
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>
```

En la versión 5.1.1 del Cisco IOS XR y posterior, el MTU y el MRU que se utiliza en el driver de la interfaz de Ethernet ha cambiado y ahora se basa en el número de etiquetas del VLA N que se configuran en los subinterfaces uces de los.

Si no se configura ningunas etiquetas del VLA N en ningunos subinterfaces, el driver MTU/MRU iguala el MTU configurado en la interfaz + 4 bytes CRC, por ejemplo $1514 + 4 = 1518$ bytes.

Si un VLA N se configura en cualesquiera subinterfaces, el driver MTU/MRU iguala el MTU+ configurado 8 bytes (1 etiqueta + CRC), por ejemplo $1514 + 8 = 1522$ bytes.

Si dos etiquetas del VLA N se configuran en cualesquiera subinterfaces, el driver MTU/MRU iguala el MTU+ configurado 12 bytes (2 etiquetas + CRC), por ejemplo $1514 + 12 = 1526$ bytes

Si QinQ con la **cualquier** palabra clave se configura para la etiqueta second-do1q el driver MTU/MRU iguala el MTU+ configurado 8 bytes (1 etiqueta + CRC), por ejemplo $1514 + 8 = 1522$ bytes.

Estos ejemplos visualizan el comportamiento en la versión 5.1.1 del Cisco IOS XR y después un 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>

```

En la mayoría de las situaciones este cambio del comportamiento en la versión 5.1.1 y posterior no debe requerir ninguna cambios a la configuración del MTU en la interfaz.

Este cambio del comportamiento puede causar los problemas en el caso de una sub-interfaz

configurada con una sola etiqueta del VLA N, pero recibe los paquetes con dos etiquetas del VLA N. En esa situación los paquetes recibidos pueden exceder el MRU en el driver de la interfaz de Ethernet. Para eliminar esa condición, el MTU de interfaz se puede aumentar en 4 bytes o la sub-interfaz configurada con dos etiquetas del VLA N.

El driver automático MTU de la interfaz de Ethernet y la configuración MRU en el comportamiento de la versión 5.1.1 es lo mismo para los 9000 Router CRS y ASR. Pero CRS un router que funciona con la versión 5.1.1 no incluye 4 el byte CRC en el valor MTU y MRU visualizado en el **resultado del controlador de la demostración**. El comportamiento de cómo está señalado no es lo mismo en medio CRS y 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)
```

La manera el MTU y el MRU se visualiza en el resultado del controlador de la demostración en el ASR 9000 cambiará en el futuro de modo que los 4 bytes de CRC no sean incluidos en el valor MTU/MRU visualizaran. Este cambio futuro se puede seguir con el Id. de bug Cisco [CSCuo93379](https://www.cisco.com/cisco/web/bugtools/bugsearch.html?bugid=93379).

Convierta la configuración cuando usted actualiza de una versión anterior que 5.1.1 para liberar 5.1.1 o más adelante

- MTU predeterminado:

Si había una interfaz principal sin ninguna sub-interfaz y sin ningún **comando mtu** en una versión anterior que la versión 5.1.1:

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

Y esta interfaz transporta el dot1q o las tramas de QinQ, después el MTU se debe configurar manualmente a "MTU el 1522" en la versión 5.1.1 y posterior:


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

Esta configuración permite que las tramas de QinQ sean transportadas como en las versiones anteriores. El valor MTU se podría configurar a 1518 si solamente se van el dot1q y no QinQ a ser transportados.

Si había subinterfaces configurados para el dot1q o QinQ, pero con el “cualquier” no se configuró la palabra clave y ningunos subinterfaces de QinQ con 2 etiquetas explícitas en una versión anterior que la versión 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
!
```

Esta configuración en la versión 5.1.1 y posterior permitirá solamente a las tramas de transporte con una etiqueta así que el MTU se debe también aumentar manualmente en 4 bytes si se van las tramas de QinQ a ser transportadas:

```
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
!
```

Si una sub-interfaz de QinQ con 2 etiquetas explícitas (que no utilice la “ninguna” palabra clave) se configura, no hay necesidad de modificar la configuración MTU cuando usted actualiza para liberar 5.1.1 y posterior:

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

Si no hay sub-interfaz del transporte L2 sino solamente las interfaces ruteadas L3, se espera que la configuración MTU haga juego en los ambos lados y no habría tramas más grandes que el MTU se transporta que. No hay necesidad de poner al día la configuración MTU cuando usted actualiza para liberar 5.1.1 y posterior.

- No valor por defecto MTU en la versión anterior que la versión 5.1.1:

Semejantemente, cuando un no valor por defecto MTU fue configurado en una versión anterior que 5.1.1 y no se configuró ninguna sub-interfaz y el dot1q o las tramas de QinQ tiene que ser transportado, después el valor configurado MTU se debe aumentar en 8 bytes cuando usted actualiza para liberar 5.1.1 o más adelante.

Versión anterior que la versión 5.1.1:

```
interface TenGigE0/1/0/19
```

```
mtu 2000
l2transport
!
```

El MTU se debe aumentar manualmente en 8 bytes cuando usted actualiza para liberar 5.1.1 y posterior:

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

El valor configurado MTU se debe también aumentar en 4 bytes si hay una sub-interfaz del dot1q y ninguna sub-interfaz de QinQ o una sub-interfaz de QinQ con la cualquier palabra clave para la etiqueta second-dot1q.

Versión anterior que la versión 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
!
```

Versión 5.1.1 y 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
!
```

Si una sub-interfaz de QinQ con 2 etiquetas explícitas (que no utilice la “ninguna” palabra clave) se configura, no hay necesidad de modificar la configuración MTU cuando usted actualiza para liberar 5.1.1 y posterior.

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

Si no hay sub-interfaz del transporte L2, sino solamente las interfaces ruteadas L3, se espera que la configuración MTU haga juego en los ambos lados y no habría tramas más grandes que el MTU se transporta que. No hay necesidad de poner al día la configuración MTU cuando usted actualiza para liberar 5.1.1 y posterior.