

Comportement de MTU sur le Cisco IOS XR et les routeurs Cisco IOS

Contenu

[Introduction](#)

[Informations générales](#)

[Configurez](#)

[Comparaison de Cisco IOS et de Logiciel Cisco IOS XR](#)

[Interfaces L3 conduites](#)

[MTU par défaut](#)

[MTU de Non-par défaut](#)

[Sous-interfaces L3 conduites](#)

[Interface du L2VPN L2](#)

[EVC \(ASR9000\)](#)

[Non EVC \(XR 12000 et CRS\)](#)

[Configuration automatique de MTU et MRU de gestionnaire d'interface Ethernet](#)

[Convertissez la configuration quand vous améliorez d'une version plus tôt que relâchent 5.1.1 pour relâcher 5.1.1 ou plus tard](#)

Introduction

Ce document décrit des comportements de Maximum Transmission Unit (MTU) sur des Routeurs du Cisco IOS® XR et compare ces comportements aux routeurs Cisco IOS. Lui discute également des mtu sur les interfaces conduites de la couche 3 (L3) et les interfaces L2 de la couche 2 VPN (L2VPN) qui utilisent les modèles de la connexion virtuelle (EVC) et du non-EVC d'Ethernets. Ce document décrit également d'importantes modifications à la façon dont le MTU et le maximum de gestionnaire d'interface d'Ethernets reçoivent l'unité (MRU) sont automatiquement configurés dans la version 5.1.1 et ultérieures.

[Informations générales](#)

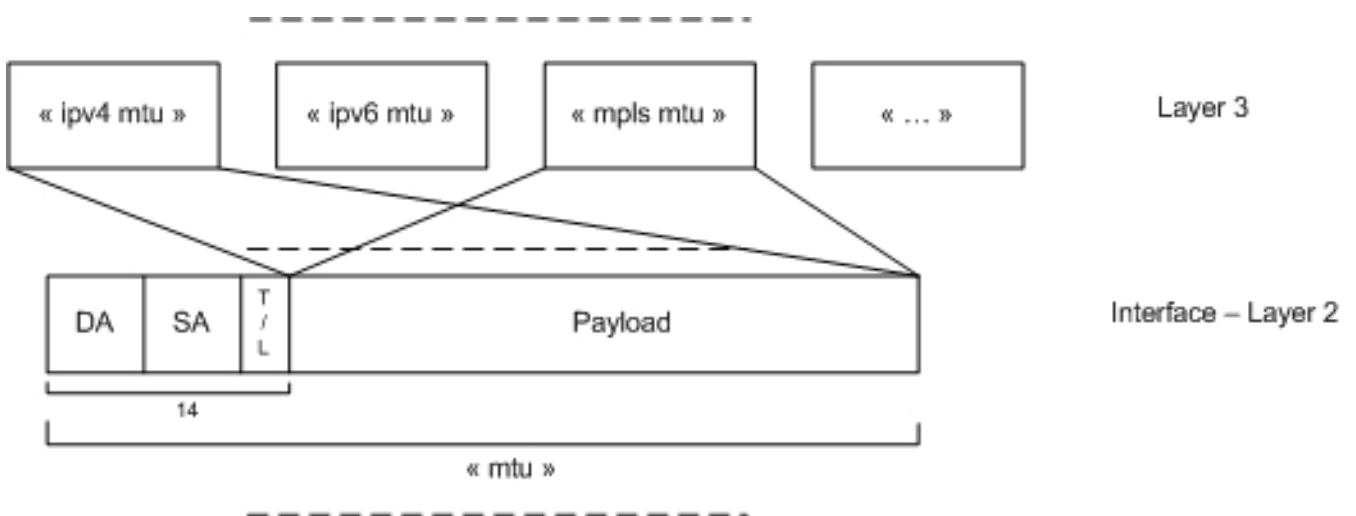
Dans le réseau informatique, le MTU d'un protocole de transmissions d'une couche définit la taille, dans les octets, du plus grand Protocol Data Unit qu'on laisse la couche pour transmettre plus d'une interface. Un paramètre de MTU est associé avec chaque interface, couche, et protocole.

Les caractéristiques de MTU dans le Logiciel Cisco IOS XR sont :

- **Le config** et les **commandes show de MTU**, à L2 et à L3, incluent la taille d'en-tête de leur couche correspondante. Par exemple, la commande de **mtu** qui configure le MTU L2 inclut 14 octets pour une interface Ethernet (sans dot1q), ou 4 octets pour le Protocole point à point

(PPP) ou le High-Level Data Link Control (HDLC). La commande d'**ipv4 mtu** inclut 20 octets de l'en-tête d'ipv4.

- Le MTU d'une couche plus élevée doit entrer dans la charge utile de la couche inférieure. Par exemple, si l'interface MTU d'une interface Ethernet non-dot1q est le par défaut de 1514 octets, puis les protocoles de couche plus élevée tels que le Commutation multiprotocole par étiquette (MPLS) peuvent avoir un MTU maximal de 1500 octets sur cette interface. Ceci signifie que vous pouvez adapter seulement une trame de 1500 octets MPLS (étiquettes y compris) à l'intérieur de la trame Ethernet. Vous ne pouvez pas configurer un MPLS MTU de 1508 octets sur cette interface si vous voulez permettre deux balises MPLS sur un paquet d'ipv4 de 1500 octets. Afin de transmettre une trame de 1508 octets MPLS sur une interface Ethernet, l'interface MTU doit être grimpée jusqu'à 1522, ou à valeur supérieure, afin de s'assurer que la charge utile de l'interface L2 est assez grande pour porter la trame MPLS.



- En logiciel classique de Cisco IOS (pas le Logiciel Cisco IOS XR), la commande de **mtu d'interface** configure la taille de la charge utile L2, mais n'inclut pas l'en-tête L2. C'est différent du Logiciel Cisco IOS XR qui inclut le temps système L2 et L3 dans la commande de **mtu d'interface**. Le MTU L3 commande, comme dans le cas de la commande d'**ipv4 mtu**, configurent la taille de paquet maximale de ce protocole qui inclut l'en-tête L3. C'est semblable dans le cas de Logiciel Cisco IOS XR.
- L'interface MTU par défaut dans le Logiciel Cisco IOS XR doit permettre le transport 1500 d'un paquet de l'octet L3. Par conséquent, le MTU par défaut est de 1514 octets pour une interface Ethernet principale et de 1504 octets pour une interface série.

Le reste de ce document illustre des caractéristiques de MTU, compare le comportement de Cisco IOS et de Logiciel Cisco IOS XR, et donne des exemples pour ces types d'interfaces :

- Interfaces L3 conduites
- Sous-interfaces L3 conduites
- Interfaces du L2VPN L2

Configurez

Remarque: Utilisez l'[Outil de recherche de commande](#) (clients [enregistrés](#) seulement) pour

obtenir plus d'informations sur les commandes utilisées dans cette section.

Remarque: [L'Output Interpreter Tool](#) (clients [enregistrés](#) seulement) prend en charge certaines **commandes show**. Utilisez l'Output Interpreter Tool afin de visualiser une analyse de sortie de commande show.

Comparaison de Cisco IOS et de Logiciel Cisco IOS XR

Cette section compare le comportement de Cisco IOS et de Logiciel Cisco IOS XR concernant des caractéristiques de MTU.

En logiciel de Cisco IOS, la commande de **mtu** et les **commandes show** correspondantes n'incluent pas l'en-tête L2. Employez la commande de **mtu** afin de configurer la charge utile L2 à la taille maximale pour les paquets L3, y compris l'en-tête L3.

C'est différent du Logiciel Cisco IOS XR, où la commande de **mtu** inclut l'en-tête L2 (14 octets pour des Ethernets ou 4 octets pour PPP/HDLC).

Si un routeur Cisco IOS est configuré avec le **mtu X** et est connecté à un routeur de Cisco IOS XR, alors l'interface correspondante sur le routeur de Cisco IOS XR devrait être configurée avec le **mtu $x+14$** pour des interfaces Ethernet, ou le **mtu $x+4$** pour des interfaces série.

Le Cisco IOS et le Logiciel Cisco IOS XR ont la même signification pour l'**ipv4 mtu**, l'**ipv6 mtu** et les commandes de **mpls mtu** ; ils doivent être configurés avec les mêmes valeurs.

En conséquence, c'est la configuration en logiciel de Cisco IOS sur une interface Ethernet :

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

La configuration correspondante sur le voisin de Logiciel Cisco IOS XR est :

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

Interfaces L3 conduites

Les valeurs de MTU doivent être identiques sur tous les périphériques connectés à un réseau L2. Autrement, ces symptômes pourraient être signalés :

- Des contiguïtés de Protocole IS-IS (Intermediate System-to-Intermediate System) ne sont pas soulevées. Par défaut, l'IS-IS utilise le hello-padding ; donc, des hellos pourraient être caractérisés comme trames géantes et pourraient être relâchés quand un routeur a une valeur de MTU qui est inférieure que les valeurs aux autres Routeurs.
- Les contiguïtés de Protocole OSPF (Open Shortest Path First) sont bloqué dans Exstart ou permutent l'état, parce que de grands paquets du descripteur de base de données (DBD) pourraient être caractérisés comme trames géantes et pourraient être lâchés. Quand les paquets sont reçus sur un routeur avec une valeur plus basse de MTU, les bases de données ne sont pas synchronisées.

- Le trafic de données est caractérisé comme trames géantes et est abandonné quand il est reçu sur un périphérique avec une valeur de MTU qui est inférieure que celle au périphérique transmetteur.
- Il y a de bas débit quand les grands paquets obtiennent relâché. En cas de découverte de MTU de chemin, la session TCP peut récupérer quand de grands paquets sont lâchés, mais ceci affecte le débit.

MTU par défaut

Cette section analyse le MTU par défaut d'une interface conduite quand la commande de `mtu` n'est pas configurée :

```
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#
```

Dans cet exemple, l'interface MTU du par défaut L2 est de 1514 octets et inclut 14 octets d'en-tête

Ethernet. Les 14 octets sont rendus compte par 6 octets d'adresse MAC de destination, 6 octets de l'adresse MAC source, et 2 octets de type ou de longueur. Ceci n'inclut pas le préambule, le délimiteur de trame, 4 octets du Frame Check Sequence (FCS), et l'écart d'inter-trame. Pour un PPP ou une trame HDLC, 4 octets de l'en-tête L2 sont expliqués ; ainsi l'interface MTU par défaut est de 1504 octets.

Les protocoles de l'enfant L3 héritent de leur MTU de la charge utile du MTU de parent. Quand vous soustrayez 14 octets d'une en-tête L2 d'un MTU L2 de 1514 octets, vous avez une charge utile L2 de 1500 octets. Ceci devient le MTU pour les protocoles L3. L'ipv4, l'IPv6, le MPLS, et le service réseau sans connexion (CLNS) héritent du MTU de l'octet this1500. En conséquence, une interface Ethernet de Cisco IOS XR, par défaut, peut transporter 1500 un paquet de l'octet L3 qui est identique que le default sur une interface Ethernet de Cisco IOS.

MTU de Non-par défaut

Cette section affiche comment configurer un **mpls mtu de 1508** afin d'envoyer un paquet de 1500 octets avec deux balises MPLS de 4 octets chacun d'ipv4, sur le paquet :

```
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#
```

Bien que la commande du **mpls mtu 1508** soit commise, elle n'est pas appliquée, parce que le MPLS a toujours un MTU de 1500 octets dans la **commande show**. C'est parce que les protocoles de l'enfant L3 ne peuvent pas avoir un MTU plus grand que la charge utile de leur interface du parent L2.

Afin de permettre deux étiquettes sur des 1500 paquets IP d'octet, vous devez :

- Configurez une interface MTU L2 de 1522 octets, de sorte que tous les protocoles d'enfant (MPLS y compris) héritent d'un MTU de 1508 octets (1522 - 14 = 1508).
- Ramenez le MTU des protocoles L3 à 1500 octets, de sorte qu'on permette seulement au MPLS pour dépasser 1500 octets.

```
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#
```

Cette configuration vous permet d'envoyer des paquets d'ipv4 et d'IPv6 de 1500 octets et des paquets MPLS de 1508 octets (un paquet de 1500 octets avec deux balises sur le dessus).

Sous-interfaces L3 conduites

Ces caractéristiques appliquent les sous-interfaces L3 conduites.

Un MTU conduit de sous-interface hérite du MTU de son interface principale de parent ; ajoutez 4 octets pour chaque balise VLAN configurée sur la sous-interface. Ainsi, il y a 4 octets pour une sous-interface dot1q et 8 octets pour un 802.1Q d'IEEE perçant un tunnel la sous-interface (de QinQ).

En conséquence, les paquets L3 de la même taille peuvent être expédiés sur l'interface principale et la sous-interface.

La commande de **mtu** peut être configurée sous la sous-interface, mais elle est appliquée seulement si elle est inférieure ou égale au MTU qui est hérité de l'interface principale.

C'est un exemple où le MTU de l'interface principale est de 2000 octets :

```
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#
```

Dans les **commandes show**, le MTU de la sous-interface est 2004 ; ajoutez 4 octets au MTU de l'interface principale parce qu'il y a une balise dot1q configurée sous la sous-interface.

Cependant, le MTU des paquets d'ipv4 et d'IPv6 sont toujours identique que celui de l'interface

principale (1986). C'est parce que le MTU des protocoles L3 est maintenant calculé en tant que :
2004 - 14 - 4 = 1986.

La commande de **mtu** peut être configurée sous la sous-interface, mais le MTU configuré est appliqué seulement s'il est inférieur ou égal au MTU qui est hérité de l'interface principale (4 octets plus grand que le MTU de l'interface principale).

Quand le MTU de la sous-interface qui est plus grande que le MTU hérité, il n'est pas appliqué, comme affiché ici :

```
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#
```

Ainsi, vous pouvez employer seulement la commande de **mtu** afin de diminuer la valeur de MTU héritée de l'interface principale.

De même, vous pouvez également employer les commandes de MTU des protocoles L3 (ipv4, IPv6, MPLS) afin de diminuer la valeur du MTU L3 hérité de la charge utile de la sous-interface L2. Le MTU du protocole L3 ne le prend pas effet quand il est configuré à une valeur qui ne s'adapte pas dans la charge utile du MTU L2.

Interface du L2VPN L2

Le MTU pour un L2VPN est important parce que le protocole de distribution d'étiquette (LDP) n'apporte pas un pseudowire (picowatt) vers le haut de quand les mtu sur les circuits de connexion sur chaque côté d'un picowatt ne sont pas identiques.

Voici une **commande show** qui illustre qu'un L2VPN picowatt reste vers le bas quand il y a une non-concordance 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#
```

Dans cet exemple, notez que les Provider Edge de L2VPN MPLS (siège potentiel d'explosion) sur chaque côté doivent signaler la même valeur de MTU afin d'évoquer le picowatt.

Le MTU signalé par MPLS LDP n'inclut pas le temps système L2. C'est différent de la configuration d'interface et des **commandes show XR** qui incluent le temps système L2. Le MTU sur la sous-interface est de 2018 octets (comme hérité de l'interface principale de 2014 octets), mais le LDP a signalé un MTU de 2000 octets. En conséquence, il soustrait 18 octets (14 octets d'en-tête Ethernet + 4 octets de 1 balise dot1q) de l'en-tête L2.

Il est important de comprendre comment chaque périphérique calcule les valeurs de MTU des circuits de connexion afin de réparer des non-concordances de MTU. Ceci dépend des paramètres tels que le constructeur, la plate-forme, la version de logiciel, et la configuration.

EVC (ASR9000)

L'agrégation de gamme 9000 de Cisco ASR entretient le routeur utilise le modèle d'infrastructure EVC, qui permet le VLAN flexible s'assortissant sur des interfaces et des sous-interfaces du

L2VPN L2.

Les interfaces du L2VPN L2 EVC ont ces caractéristiques :

- Ils permettent la configuration d'un ou plusieurs balises avec la commande d'**encapsulation**.
- Par défaut et avec juste la commande d'**encapsulation**, des balises sont préservées et transportées au-dessus de PWs. En conséquence, vous n'avez pas besoin d'éliminer des balises par défaut, pendant que vous devez faire sur des Plateformes de non-EVC.
- Utilisez la commande de **réécriture** quand vous décidez de sauter les balises entrantes ou de pousser quelques balises supplémentaires sur la trame entrante.

Afin de calculer le MTU de sous-interface, prendre le MTU d'interface principale (le par défaut ou celui configuré manuellement sous l'interface principale), et ajouter 4 octets pour chaque balise VLAN configurée avec l'**encapsulation** commandez. Voir les [commandes spécifiques d'encapsulation EFP](#).

Quand il y a une commande de **mtu** sous la sous-interface, elle la prend effet seulement si elle est inférieure au MTU calculé. La commande de **réécriture** n'influence pas le MTU de la sous-interface.

Voici un exemple :

```
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)
```

Dans cet exemple, le MTU dans l'interface principale est de 2014 octets ; ajoutez 8 octets parce qu'il y a deux balises configurées sous la sous-interface.

Si vous configurez le **mtu 2026** octets sous la sous-interface, elle n'est pas appliquée parce qu'elle est plus grande que le MTU de la sous-interface héritée de l'interface principale (2022). En conséquence, vous pouvez configurer seulement octets inférieurs d'un MTU de sous-interface les que 2022.

Basé sur ce MTU de sous-interface, calculez le MTU de la charge utile MPLS LDP qui est signalée au voisin, et assurez-vous qu'elle est identique à celui calculé par le PE de L2VPN distant. C'est où la commande de **réécriture** entre dans le jeu.

Afin de calculer le MTU de la charge utile MPLS LDP, prenez le MTU de la sous-interface, puis :

1. Soustrayez 14 octets pour l'en-tête Ethernet.
2. Soustrayez 4 octets pour chaque balise sautée dans la commande de **réécriture** configurée sous la sous-interface.
3. Ajoutez 4 octets pour chaque balise enfoncée la commande de **réécriture** configurée sous la sous-interface.

C'est le même exemple avec la configuration de QinQ sur la yole 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)
```

C'est le calcul pour le MTU de la charge utile MPLS LDP :

1. Valeur de MTU du MTU de sous-interface : 2022 octets
2. Soustrayez 14 octets d'en-tête Ethernet : $2022 - 14 = 2008$ octets
3. Soustrayez 4 octets pour chaque balise sautée dans la commande de **réécriture** : $2008 - 4 * 2 = 2000$

Assurez-vous que le côté distant annonce une charge utile MPLS LDP de 2000 octets. Autrement, ajustez le local ou taille distante de MTU du circuit de connexion (courant alternatif) ainsi qu'ils appartiennent.

```
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
```

Les Ethernets spécifiques circulent des commandes d'encapsulation du point (EFP)

Ces encapsulations comptent en tant que balises zéro assorties, ainsi elles n'augmentent pas le MTU de sous-interface :

- encapsulation untagged
- encapsulation default

Ces modificateurs d'encapsulation n'affectent pas le nombre de balises exigées afin de calculer le MTU de sous-interface :

- indigène
- charge utile-ethertype
- précis

- COS
- source-MAC d'entrée ou destination-MAC d'entrée

l'encapsulation [dot1q|dot1ad] priorité-a étiqueté des comptes en tant qu'apparier une balise simple.

Le « aucun » mot clé utilisé à mesure que la correspondance de balise les plus secrets n'augmente le MTU de sous-interface.

- **l'encapsulation dot1q** n'augmente pas le MTU de sous-interface.
- **l'encapsulation dot1ad 10 dot1q** est rendue compte comme une balise ; il augmente le MTU de sous-interface de 4 octets.
- **l'encapsulation dot1ad n'importe quel dot1q 7** est rendue compte en tant que deux balises ; il augmente le MTU de sous-interface de 8 octets.

Les plages des IDs de VLAN incrémentent le MTU de sous-interface :

- **l'encapsulation dot1q 10-100** est rendue compte comme une balise ; il augmente le MTU de sous-interface de 4 octets.

Le temps système de MTU d'encapsulation d'un EFP qui est une correspondance disjonctive est traité comme MTU de son élément plus élevé.

- **l'encapsulation dot1q 10-100, non-marquée** est rendue compte car une balise parce que la plage 10 -100 est l'élément le plus élevé.

Non EVC (XR 12000 et CRS)

Les Routeurs comme le Routeur de la gamme Cisco XR 12000 et le système de routage de transporteur (CRS) utilisent la configuration traditionnelle pour le VLAN s'assortissant sur des sous-interfaces. Ces caractéristiques s'appliquent aux interfaces du L2VPN L2 sur des CRS et sur les Routeurs XR 12000 qui ne suivent pas le modèle EVC :

- Sur des Plateformes de non-EVC, les balises entrantes dot1q ou dot1ad sont automatiquement éliminées quand elles sont reçues sur une sous-interface de l2transport.
- Quand vous calculez la taille de charge utile pour MPLS LDP pour signaler, soustrayez la taille des balises du MTU de la sous-interface, comme vu dans la **commande d'interface d'exposition**.
- C'est semblable dans le cas d'une sous-interface conduite.
- La sous-interface hérite de son MTU de l'interface principale ; ajoutez les 4 octets pour chaque balise au MTU de l'interface principale afin de calculer le MTU de la sous-interface. Par exemple, si une sous-interface de QinQ a 2 balises dot1q, la sous-interface, par défaut, a un MTU qui est 8 octets plus grand que le MTU de l'interface principale.
- Vous pouvez également utiliser la commande de **mtu** sous la sous-interface, mais elle est utilisée pour réduire seulement le MTU de la sous-interface, qui est héritée du MTU de l'interface principale.

Voici plusieurs exemples qui montrent ces caractéristiques.

Cet exemple affiche comment une sous-interface de non-EVC est configurée :

```
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#
```

Les Plateformes de non-EVC utilisent le **dot1q vlan** ou les commandes de **VLAN dot1ad** au lieu de l'**encapsulation et réécrivent des** commandes des Plateformes EVC (ASR9000).

Si vous ne configurez pas un MTU explicitement sur la canalisation ou la sous-interface, alors 1500 un paquet de l'octet L3 peut être reçu par défaut :

```
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#
```

Le MTU de sous-interface est calculé du MTU d'interface principale (1514) ; ajoutez 4 octets pour chaque balise dot1q. Puisqu'il y a une balise configurée sur la sous-interface avec la commande du **dot1q vlan 201**, ajoutez 4 octets à 1514 pour un MTU de 1518 octets.

Le MTU correspondant de charge utile dans MPLS LDP est de 1500 octets, puisque les 14 octets de l'en-tête Ethernet ne sont pas comptés et l'une balise dot1q est sautée automatiquement par la plate-forme de non-EVC quand elle va au-dessus du 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
```

Si vous augmentez le MTU de l'interface principale à 2014 octets, le MTU de la sous-interface est augmenté en conséquence :

```
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
```

Ainsi, afin de calculer le MTU MPLS LDP, soustraire 14 octets d'en-tête Ethernet, et ajouter 4

octets pour chaque balise configurée sous la sous-interface.

Configuration automatique de MTU et MRU de gestionnaire d'interface Ethernet

Sur des interfaces Ethernet le gestionnaire d'interface est configuré avec un MTU et un MRU qui est basé sur la configuration d'interface MTU.

Le MTU et le MRU configurés sur le gestionnaire d'interface Ethernet peuvent être vus avec le **<interface> de show controller toute la commande.**

Dans des releases plus tôt que la version 5.1.1 de Cisco IOS XR, le MTU et les MRU sur le gestionnaire d'interface Ethernet ont été automatiquement configurés basés sur la configuration de MTU de Cisco IOS XR sur l'interface.

Le MTU/MRU configuré sur le gestionnaire d'Ethernets a été simplement basé sur le MTU configuré + 12 octets pour l'ajout de 2 balises d'Ethernets et du champ CRC. Les 12 octets ont été ajoutés au gestionnaire MTU/MRU d'Ethernets indépendamment de s'il y avait des balises VLAN configurées sur les sous-interfaces.

Un exemple avec toutes les versions de Cisco IOS XR plus tôt que la version 5.1.1 de Cisco IOS XR et un MTU par défaut de 1514 sur une interface ASR 9000 est affiché ici :

```
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>
```

Dans la version 5.1.1 et ultérieures de Cisco IOS XR, le MTU et le MRU qui est utilisé sur le gestionnaire d'interface d'Ethernets a changé et est maintenant basé sur le nombre d'étiquettes VLAN qui sont configurées sur les sous-interfaces l'un des.

Si aucune balise VLAN n'est configurée sur aucune sous-interface, le gestionnaire MTU/MRU égale le MTU configuré sur l'interface + 4 octets de CRC, par exemple 1514 + 4 = 1518 octets.

Si un VLAN est configuré sur n'importe quelles sous-interfaces, le gestionnaire MTU/MRU égale le

MTU configuré + 8 octets (1 balise + CRC), par exemple 1514 + 8 = 1522 octets.

Si deux balises VLAN sont configurées sur n'importe quelles sous-interfaces, le gestionnaire MTU/MRU égale le MTU configuré + 12 octets (2 balises + CRC), par exemple 1514 + 12 = 1526 octets

Si QinQ avec le **n'importe quel** mot clé est configuré pour la balise second-dot1q le gestionnaire MTU/MRU égale le MTU configuré + 8 octets (1 balise + CRC), par exemple 1514 + 8 = 1522 octets.

Ces exemples affichent le comportement dans la version 5.1.1 de Cisco IOS XR et plus tard 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>

Dans la plupart des situations ce changement de comportement de version 5.1.1 et ultérieures ne devrait exiger aucune modifications à la configuration du MTU sur l'interface.

Cette modification de comportement peut poser des problèmes dans le cas d'une sous-interface configurée avec une balise de VLAN simple, mais reçoit des paquets avec deux balises VLAN. Dans cette situation les paquets reçus peuvent dépasser le MRU sur le gestionnaire d'interface Ethernet. Afin d'éliminer cette condition, l'interface MTU peut être augmentée par 4 octets ou la sous-interface configurée avec deux balises VLAN.

La configuration automatique de MTU et MRU de gestionnaire d'interface Ethernet dans le comportement de la version 5.1.1 est identique pour des Routeurs CRS et ASR 9000. Mais un routeur CRS qui exécute la version 5.1.1 n'inclut pas le CRC de 4 octets en valeur de MTU et MRU affichée dans la sortie de **show controller**. Le comportement de la façon dont il est signalé n'est pas identique entre les CRS et l'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 manière le MTU et le MRU est affichée dans la sortie de show controller sur l'ASR 9000 changera à l'avenir de sorte que les 4 octets de CRC ne soient pas inclus en valeur MTU/MRU aient affiché. Cette future modification peut être dépiquée avec l'ID de bogue Cisco [CSCuo93379](https://tools.cisco.com/bugcenter/bug/?bugID=CSCuo93379).

Convertissez la configuration quand vous améliorez d'une version plus tôt que relâchent 5.1.1 pour relâcher 5.1.1 ou plus tard

- MTU par défaut :

S'il y avait une interface principale sans n'importe quelle sous-interface et sans n'importe quelle commande de **mtu** dans une version plus tôt que la version 5.1.1 :

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

Et cette interface transporte dot1q ou trames de QinQ, puis le MTU devrait être manuellement configuré au « mtu 1522" dans la version 5.1.1 et ultérieures :

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

Cette configuration permet des trames de QinQ à transporter comme dans les versions antérieures. La valeur de MTU pourrait être configurée à 1518 si seulement dot1q et pas QinQ doivent être transportés.

S'il y avait des sous-interfaces configurées pour dot1q ou QinQ, mais avec le « tout » le mot clé et aucune sous-interface de QinQ avec 2 balises explicites n'ont été configurés dans une version plus tôt que la version 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
!
```

Cette configuration dans la version 5.1.1 et ultérieures autorisera seulement aux trames de transport avec une étiquette ainsi le MTU devrait également être augmenté manuellement de 4 octets si des trames de QinQ doivent être transportées :

```
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 une sous-interface de QinQ avec 2 étiquettes explicites (qui n'utilisent le « aucun » mot clé) est configurée, il n'y a aucun besoin de modifier la configuration de MTU quand vous améliorez à la version 5.1.1 et ultérieures :

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

S'il n'y a aucune sous-interface de l2transport mais seulement interfaces conduites par L3, on s'attend à ce que la configuration de MTU s'assortisse sur les deux côtés et il n'y aurait pas des trames plus grandes que le MTU qui est transporté. Il n'y a aucun besoin de mettre à jour la configuration de MTU quand vous améliorez à la version 5.1.1 et ultérieures.

- MTU de Non-par défaut dans la version plus tôt que la version 5.1.1 :

De même, quand un MTU de non-par défaut été configuré dans une version plus tôt que relâchent 5.1.1 et aucune sous-interface a n'a été configurée et dot1q ou trames de QinQ doivent être transportés, puis la valeur configurée de MTU devrait être augmentée de 8 octets quand vous améliorez pour relâcher 5.1.1 ou plus tard.

Version plus tôt que la version 5.1.1 :

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

Le MTU devrait être manuellement augmenté de 8 octets quand vous améliorez à la version 5.1.1 et ultérieures :

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

La valeur configurée de MTU devrait également être augmentée de 4 octets s'il y a une sous-interface dot1q et aucune sous-interface de QinQ ou une sous-interface de QinQ avec le n'importe quel mot clé pour la balise second-dot1q.

Version plus tôt que la version 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  
!
```

Version 5.1.1 et ultérieures :

```
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 une sous-interface de QinQ avec 2 étiquettes explicites (qui n'utilisent le « aucun » mot clé) est configurée, il n'y a aucun besoin de modifier la configuration de MTU quand vous améliorez à la version 5.1.1 et ultérieures.

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

S'il n'y a aucune sous-interface de l2transport, mais seulement L3 conduit relie, on s'attend à ce que la configuration de MTU s'assortisse sur les deux côtés et il n'y aurait pas des trames plus grandes que le MTU qui est transporté. Il n'y a aucun besoin de mettre à jour la configuration de MTU quand vous améliorez à la version 5.1.1 et ultérieures.