

Comportement MTU sur Cisco IOS XR et les routeurs Cisco IOS

Table des matières

[Introduction](#)

[Informations générales](#)

[Configurer](#)

[Comparaison des logiciels Cisco IOS et Cisco IOS XR](#)

[Interfaces C3 routées](#)

[MTU par défaut](#)

[MTU non par défaut](#)

[Sous-interfaces L3 routées](#)

[Interface L2VPN L2](#)

[EVC \(ASR9000\)](#)

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

[Configuration MTU et MRU du pilote d'interface Ethernet automatique](#)

[Convertir la configuration lors de la mise à niveau d'une version antérieure à la version 5.1.1 vers la version 5.1.1 ou ultérieure](#)

Introduction

Ce document décrit les comportements d'unité de transmission maximale (MTU) sur les routeurs Cisco IOS[®] XR et compare ces comportements aux routeurs Cisco IOS. Il traite également des MTU sur les interfaces routées de couche 3 (L3) et les interfaces de couche 2 (L2VPN) de réseau privé virtuel de couche 2 qui utilisent à la fois les modèles Ethernet Virtual Connection (EVC) et non-EVC. Ce document décrit également les modifications importantes apportées à la configuration automatique de la MTU du pilote d'interface Ethernet et de l'unité de réception maximale (MRU) dans les versions 5.1.1 et ultérieures.

Informations générales

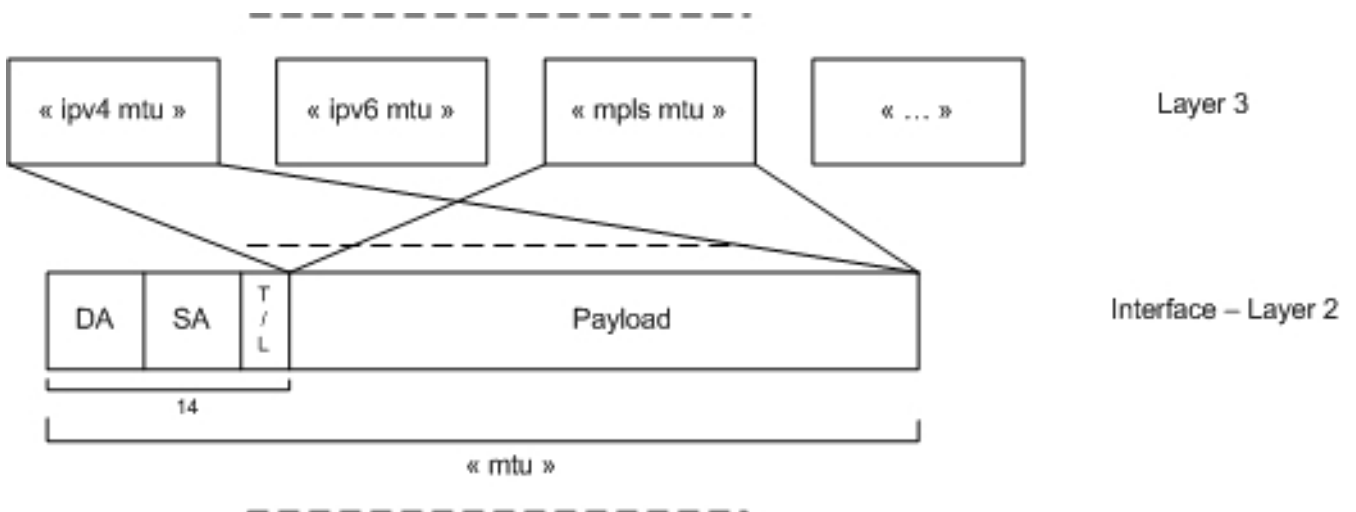
Dans un réseau informatique, le MTU d'un protocole de communication d'une couche définit la taille, en octets, de la plus grande unité de données de protocole que la couche est autorisée à transmettre sur une interface. Un paramètre MTU est associé à chaque interface, couche et protocole.

Les caractéristiques MTU du logiciel Cisco IOS XR sont les suivantes :

- Les commandes **config** et **show** de l'unité de transmission maximale, en L2 et en L3, incluent la taille d'en-tête de leur couche correspondante. Par exemple, la commande **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 contrôle de liaison de données de haut niveau (HDLC). La commande **ipv4 mtu** inclut 20 octets de l'en-tête IPv4.

- Le MTU d'une couche supérieure doit tenir dans la charge utile de la couche inférieure. Par exemple, si le MTU d'interface d'une interface Ethernet autre que dot1q est de 1 514 octets par défaut, les protocoles de couche supérieure tels que MPLS (Multiprotocol Label Switching) peuvent avoir un MTU maximal de 1 500 octets sur cette interface. Cela signifie que vous ne pouvez insérer qu'une trame MPLS de 1 500 octets (y compris les étiquettes) à l'intérieur de la trame Ethernet. Vous ne pouvez pas configurer un MTU MPLS de 1 508 octets sur cette interface si vous souhaitez autoriser deux balises MPLS au-dessus d'un paquet IPv4 de 1 500 octets. Afin de transmettre une trame MPLS de 1 508 octets sur une interface Ethernet, la valeur MTU de l'interface doit être augmentée à 1 522, ou une valeur supérieure, afin de s'assurer que la charge utile de l'interface L2 est suffisamment grande pour transporter la trame MPLS.



- Dans le logiciel Cisco IOS classique (pas le logiciel Cisco IOS XR), la commande interface **mtu** configure la taille de la charge utile L2, mais n'inclut pas l'en-tête L2. Il est différent du logiciel Cisco IOS XR qui inclut à la fois la surcharge L2 et L3 dans la commande **interface mtu**. Les commandes L3 MTU, comme dans le cas de la commande **ipv4 mtu**, configurent la taille de paquet maximale de ce protocole qui inclut l'en-tête L3. Cela est similaire au cas du logiciel Cisco IOS XR.
- Le MTU d'interface par défaut du logiciel Cisco IOS XR doit permettre le transport d'un paquet L3 de 1 500 octets. Par conséquent, le MTU par défaut est de 1 514 octets pour une interface Ethernet principale et de 1 504 octets pour une interface série.

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

- Interfaces C3 routées
- Sous-interfaces L3 routées
- Interfaces L2VPN L2

Configurer

Remarque : utilisez l'[outil de recherche de commandes](#) (clients [enregistrés](#) uniquement) afin d'obtenir plus d'informations sur les commandes utilisées dans cette section.

Remarque : l'[outil Output Interpreter Tool](#) (clients [enregistrés](#) uniquement) prend en charge certaines commandes **show**. Utilisez l'Outil d'interprétation de sortie afin de visualiser une analyse de commande d'affichage de sortie .

Comparaison des logiciels Cisco IOS et Cisco IOS XR

Cette section compare le comportement des logiciels Cisco IOS et Cisco IOS XR par rapport aux caractéristiques MTU.

Dans le logiciel Cisco IOS, la commande **mtu** et les commandes **show** correspondantes n'incluent pas l'en-tête L2. Utilisez la commande **mtu** afin de configurer la charge utile de couche 2 à la taille maximale pour les paquets de couche 3, y compris l'en-tête de couche 3.

Il est différent du logiciel Cisco IOS XR, où la commande **mtu** inclut l'en-tête L2 (14 octets pour Ethernet ou 4 octets pour PPP/HDLC).

Si un routeur Cisco IOS est configuré avec $mtu\ x$ et est connecté à un routeur Cisco IOS XR, l'interface correspondante sur le routeur Cisco IOS XR doit être configurée avec $mtu\ x+14$ pour les interfaces Ethernet ou $mtu\ x+4$ pour les interfaces série.

Les logiciels Cisco IOS et Cisco IOS XR ont la même signification pour les commandes **ipv4 mtu**, **ipv6 mtu** et **mpls mtu** ; ils doivent être configurés avec les mêmes valeurs.

Par conséquent, voici la configuration de la plate-forme logicielle Cisco IOS sur une interface Ethernet :

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

La configuration correspondante sur le voisin du logiciel Cisco IOS XR est la suivante :

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

Interfaces C3 routées

Les valeurs MTU doivent être identiques sur tous les périphériques connectés à un réseau L2. Sinon, ces symptômes peuvent être rapportés :

- Les contiguïtés IS-IS (Intermediate System-to-Intermediate System) ne s'affichent pas. Par défaut, IS-IS utilise le remplissage Hello. Par conséquent, les paquets Hello peuvent être caractérisés comme des paquets géants et peuvent être supprimés lorsqu'un routeur a une valeur MTU inférieure aux valeurs des autres routeurs.
- Les contiguïtés OSPF (Open Shortest Path First) sont bloquées dans l'état Exstart ou

Exchange, car les paquets DBD (Large Database Descriptor) peuvent être caractérisés comme des paquets géants et peuvent être abandonnés. Lorsque les paquets sont reçus sur un routeur avec une valeur MTU inférieure, les bases de données ne sont pas synchronisées.

- Le trafic de données est considéré comme géant et est abandonné lorsqu'il est reçu sur un périphérique avec une valeur MTU inférieure à celle du périphérique émetteur.
- Le débit est faible lorsque des paquets volumineux sont abandonnés. En cas de découverte de MTU de chemin, la session TCP peut récupérer lorsque des paquets volumineux sont abandonnés, mais cela affecte le débit.

MTU par défaut

Cette section analyse le MTU par défaut d'une interface routée quand la commande `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, le MTU de l'interface L2 par défaut est de 1 514 octets et inclut 14 octets d'en-tête Ethernet. Les 14 octets sont représentés par 6 octets d'adresse MAC de destination, 6 octets d'adresse MAC source et 2 octets de type ou de longueur. Ceci n'inclut pas le préambule, le délimiteur de trame, 4 octets de séquence de contrôle de trame (FCS) et l'intervalle entre les trames. Pour une trame PPP ou HDLC, 4 octets de l'en-tête L2 sont pris en compte ; le MTU de l'interface par défaut est donc de 1 504 octets.

Les protocoles enfants de couche 3 héritent leur MTU de la charge utile du MTU parent. Lorsque vous soustrayez 14 octets d'un en-tête L2 d'une valeur MTU L2 de 1 514 octets, vous obtenez une charge utile L2 de 1 500 octets. Il devient le MTU pour les protocoles L3. IPv4, IPv6, MPLS et CLNS (Connectionless Network Service) héritent de cette MTU de 1 500 octets. Par conséquent, une interface Ethernet XR de Cisco IOS, par défaut, peut transporter un paquet L3 de 1 500 octets qui est identique à l'interface defaultIf sur une interface Ethernet de Cisco IOS.

MTU non par défaut

Cette section montre comment configurer un **mtu mpls** de 1508 afin d'envoyer un paquet IPv4 de 1500 octets avec deux balises MPLS de 4 octets chacune, au-dessus du 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 **mpls mtu 1508** soit validée, elle n'est pas appliquée, car MPLS a toujours une MTU de 1500 octets dans la commande **show**. Cela est dû au fait que les protocoles enfants de couche 3 ne peuvent pas avoir une MTU supérieure à la charge utile de leur interface de couche 2 parente.

Pour autoriser deux étiquettes au-dessus d'un paquet IP de 1 500 octets, vous devez :

- Configurez une MTU d'interface L2 de 1 522 octets, de sorte que tous les protocoles enfants (y compris MPLS) héritent d'une MTU de 1 508 octets (1 522 - 14 = 1 508).
- Réduisez la MTU des protocoles de couche 3 à 1 500 octets, de sorte que seul MPLS soit autorisé à dépasser 1 500 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 IPv4 et IPv6 de 1 500 octets et des

paquets MPLS de 1 508 octets (un paquet de 1 500 octets avec deux balises en haut).

Sous-interfaces L3 routées

Ces caractéristiques s'appliquent aux sous-interfaces L3 routées.

Une MTU de sous-interface routée hérite de la MTU de son interface principale parente ; ajoutez 4 octets pour chaque étiquette VLAN configurée sur la sous-interface. Ainsi, il y a 4 octets pour une sous-interface dot1q et 8 octets pour une sous-interface de tunnellation IEEE 802.1Q (QinQ).

Par conséquent, les paquets L3 de même taille peuvent être transférés à la fois sur l'interface principale et sur la sous-interface.

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

Voici un exemple où la 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: 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#
```

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

Cependant, le MTU des paquets IPv4 et IPv6 reste le même que celui de l'interface principale (1986). En effet, le MTU des protocoles de couche 3 est désormais calculé comme suit : $2004 - 14 - 4 = 1986$.

La commande **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).

Lorsque le MTU de la sous-interface est plus grand que le MTU hérité, il n'est pas appliqué, comme illustré 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 utiliser seulement la commande **mtu** afin de diminuer la valeur de MTU héritée de l'interface principale.

De même, vous pouvez également utiliser les commandes MTU des protocoles L3 (IPv4, IPv6, MPLS) afin de diminuer la valeur de la MTU L3 héritée de la charge utile L2 de la sous-interface. Le MTU du protocole L3 ne prend pas effet lorsqu'il est configuré sur une valeur qui ne correspond pas à la charge utile du MTU L2.

Interface L2VPN L2

Le MTU d'un L2VPN est important, car le protocole LDP (Label Distribution Protocol) n'active pas de pseudo-fil (PW) lorsque les MTU des circuits de connexion de chaque côté d'un PW ne sont pas identiques.

Voici une commande **show** qui illustre qu'un PW L2VPN reste inactif quand il y a une incompatibilité 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 périphériques du fournisseur L2VPN MPLS de chaque côté doivent signaler la même valeur MTU afin de faire monter le PW.

Le MTU signalé par le protocole MPLS LDP n'inclut pas la surcharge de couche 2. Cette commande est différente des commandes **config** et **show** de l'interface XR qui incluent la surcharge de couche 2. La MTU sur la sous-interface est de 2018 octets (héritée de l'interface principale de 2014 octets), mais LDP a signalé une MTU de 2000 octets. Par conséquent, il soustrait 18 octets (14 octets d'en-tête Ethernet + 4 octets de balise 1 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 corriger les incohérences de MTU. Cela dépend de paramètres tels

que le fournisseur, la plate-forme, la version du logiciel et la configuration.

EVC (ASR9000)

Le routeur à services d'agrégation de la gamme Cisco ASR 9000 utilise le modèle d'infrastructure EVC, qui permet une correspondance VLAN souple sur les interfaces et sous-interfaces L2VPN L2.

Les interfaces L2VPN L2 EVC présentent les caractéristiques suivantes :

- Ils permettent de configurer une ou plusieurs balises à l'aide de la commande **encapsulation**.
- Par défaut et avec la seule commande **encapsulation**, les balises sont conservées et transportées sur des PC. Par conséquent, vous n'avez pas besoin de supprimer les balises par défaut, comme vous devez le faire sur les plates-formes non EVC.
- Utilisez la commande **rewrite** lorsque vous décidez d'afficher les balises entrantes ou de pousser des balises supplémentaires sur le dessus de la trame entrante.

Afin de calculer le MTU de la sous-interface, prenez le MTU de l'interface principale (soit le MTU par défaut soit celui configuré manuellement sous l'interface principale), et ajoutez 4 octets pour chaque étiquette VLAN configurée avec la commande **encapsulation**. Voir [Commandes d'encapsulation EFP spécifiques](#).

Lorsqu'il y a une commande **mtu** sous la sous-interface, elle prend effet seulement si elle est inférieure à la MTU calculée. La commande **rewrite** 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 de l'interface principale est de 2 014 octets ; ajoutez 8 octets car deux balises sont configurées sous la sous-interface.

Si vous configurez **mtu 2026** octets sous la sous-interface, elle n'est pas appliquée parce qu'elle est plus grande que la MTU de la sous-interface héritée de l'interface principale (2022). Par conséquent, vous ne pouvez configurer qu'une MTU de sous-interface inférieure à 2022 octets.

Sur la base de cette MTU de sous-interface, calculez la MTU de la charge utile LDP MPLS qui est signalée au voisin, et assurez-vous qu'elle est identique à celle calculée par le PE L2VPN distant. C'est là que la commande **rewrite** entre en 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 apparaissant dans la commande **rewrite** configurée sous la sous-interface.
3. Ajoutez 4 octets pour chaque balise poussée dans la commande **rewrite** configurée sous la sous-interface.

Voici le même exemple avec la configuration QinQ sur 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)
```

Il s'agit du calcul de la MTU de la charge utile MPLS LDP :

1. Valeur MTU de la MTU de la sous-interface : 2 022 octets
2. Soustraire 14 octets de l'en-tête Ethernet : $2022 - 14 = 2008$ octets
3. Soustrayez 4 octets pour chaque balise contextuelle dans le **réécrire** commande : $2008 - 4 * 2 = 2000$

Assurez-vous que le côté distant annonce une charge utile MPLS LDP de 2 000 octets. Sinon, ajustez la taille du MTU du circuit de connexion local ou distant (AC) pour qu'elle corresponde.

```
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
```

Commandes d'encapsulation EFP (Ethernet Flow Point) spécifiques

Ces encapsulations comptent comme des balises zéro correspondantes, de sorte qu'elles n'augmentent pas la MTU de la sous-interface :

- encapsulation non marquée

- **encapsulation par défaut**

Ces modificateurs d'encapsulation n'affectent pas le nombre de balises nécessaires pour calculer le MTU de la sous-interface :

- indigène
- payload-ethertype
- exact
- cos
- entrée source-mac ou entrée destination-mac

encapsulation [dot1q|dot1ad] priority-tagged compte comme correspondant à une seule balise.

Le mot clé « any » utilisé comme la correspondance de balise la plus interne n'augmente pas le MTU de la sous-interface.

- l'**encapsulation dot1q any** n'augmente pas le MTU de la sous-interface.
- **encapsulation dot1ad 10 dot1q any** est comptabilisée comme une balise ; elle augmente la MTU de la sous-interface de 4 octets.
- **encapsulation dot1ad any dot1q 7** est comptabilisée comme deux balises ; elle augmente la MTU de la sous-interface de 8 octets.

Les plages d'ID de VLAN incrémentent le MTU de la sous-interface :

- **encapsulation dot1q 10-100** est comptabilisée comme une balise ; elle augmente la MTU de la sous-interface de 4 octets.

La surcharge MTU d'encapsulation d'un EFP qui est une correspondance disjonctive est traitée comme la MTU de son élément le plus élevé.

- **encapsulation dot1q 10-100, untagged** est comptabilisé comme une balise car la plage 10 - 100 est l'élément le plus élevé.

Non EVC (XR 12000 et CRS)

Les routeurs tels que les routeurs de la gamme Cisco XR 12000 et le système de routage opérateur (CRS) utilisent la configuration traditionnelle pour la correspondance VLAN sur les sous-interfaces. Ces caractéristiques s'appliquent aux interfaces L2VPN L2 sur CRS et sur les routeurs XR 12000 qui ne suivent pas le modèle EVC :

- Sur les plates-formes non EVC, les balises dot1q ou dot1ad entrantes sont automatiquement supprimées lorsqu'elles sont reçues sur une sous-interface de transport L2.
- Lorsque vous calculez la taille de la charge utile pour le signal LDP MPLS, soustrayez la taille des balises de la MTU de la sous-interface, comme indiqué dans la commande **show interface**.
- Cela est similaire au cas d'une sous-interface routée.
- La sous-interface hérite de son MTU de l'interface principale ; ajoutez les 4 octets de chaque étiquette au MTU de l'interface principale afin de calculer le MTU de la sous-interface. Par exemple, si une sous-interface 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 **mtu** sous la sous-interface, mais elle est utilisée uniquement pour réduire la MTU de la sous-interface, qui est héritée de la MTU de l'interface principale.

Voici quelques exemples qui illustrent ces caractéristiques.

Cet exemple montre comment une sous-interface 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 plates-formes non-EVC utilisent les commandes **dot1q vlan** ou **dot1ad vlan** au lieu des commandes **encapsulation** et **rewrite** des plates-formes EVC (ASR9000).

Si vous ne configurez pas explicitement une MTU sur l'interface principale ou secondaire, un paquet L3 de 1 500 octets 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#
```

La MTU de la sous-interface est calculée à partir de la MTU de l'interface principale (1514) ; ajoutez 4 octets pour chaque étiquette dot1q. Comme une balise est configurée sur la sous-interface avec la commande **dot1q vlan 201**, ajoutez 4 octets à 1514 pour une MTU de 1518 octets.

Le MTU de charge utile correspondant dans le LDP MPLS est de 1500 octets, puisque les 14 octets de l'en-tête Ethernet ne sont pas comptés et l'étiquette one dot1q est automatiquement ajoutée par la plate-forme non-EVC lorsqu'elle passe sur le 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
```

Si vous augmentez la MTU de l'interface principale à 2014 octets, la MTU de la sous-interface est augmentée 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, soustrayez 14 octets d'en-tête Ethernet et ajoutez 4 octets pour chaque balise configurée sous la sous-interface.

Configuration MTU et MRU du pilote d'interface Ethernet automatique

Sur les interfaces Ethernet, le pilote d'interface est configuré avec une MTU et une MRU basées sur la configuration de la MTU de l'interface.

Les MTU et MRU configurés sur le pilote d'interface Ethernet peuvent être vus avec la commande **show controller <interface> all**.

Dans les versions antérieures à la version 5.1.1 de Cisco IOS XR, le MTU et le MRU sur le pilote d'interface Ethernet ont été automatiquement configurés en fonction de la configuration du MTU de Cisco IOS XR sur l'interface.

Le MTU/MRU configuré sur le pilote Ethernet était simplement basé sur le MTU configuré + 12 octets pour l'ajout de 2 balises Ethernet et le champ CRC. Les 12 octets ont été ajoutés au MTU/MRU du pilote Ethernet, qu'il y ait ou non des balises VLAN configurées sur les sous-interfaces.

Un exemple avec toutes les versions de Cisco IOS XR antérieures à Cisco IOS XR version 5.1.1 et un MTU par défaut de 1514 sur une interface ASR 9000 est montré 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 Cisco IOS XR versions 5.1.1 et ultérieures, le MTU et le MRU utilisés sur le pilote d'interface Ethernet ont changé et sont désormais basés sur le nombre de balises VLAN configurées sur l'une des sous-interfaces.

Si aucune étiquette VLAN n'est configurée sur une sous-interface, le MTU/MRU du pilote est égal au MTU configuré sur l'interface + 4 octets CRC, par exemple $1514 + 4 = 1518$ octets.

Si un VLAN est configuré sur n'importe quelle sous-interface, le MTU/MRU du pilote est égal au MTU configuré + 8 octets (1 balise + CRC), par exemple $1514 + 8 = 1522$ octets.

Si deux étiquettes VLAN sont configurées sur n'importe quelle sous-interface, le MTU/MRU du pilote est égal au MTU configuré + 12 octets (2 étiquettes + CRC), par exemple $1514 + 12 = 1526$ octets

Si QinQ avec le mot clé **any** est configuré pour la balise second-dot1q, le MTU/MRU du pilote est égal au MTU configuré + 8 octets (1 balise + CRC), par exemple $1514 + 8 = 1522$ octets.

Ces exemples affichent le comportement de Cisco IOS XR version 5.1.1 et ultérieure sur 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 cas, ce changement de comportement dans les versions 5.1.1 et ultérieures ne devrait pas nécessiter de modification de la configuration de la MTU sur l'interface.

Ce changement de comportement peut causer des problèmes dans le cas d'une sous-interface configurée avec une étiquette VLAN unique, mais reçoit des paquets avec deux étiquettes VLAN. Dans ce cas, les paquets reçus peuvent dépasser le MRU sur le pilote d'interface Ethernet. Afin d'éliminer cette condition, l'interface MTU peut être augmentée de 4 octets ou la sous-interface configurée avec deux balises VLAN.

La configuration automatique des pilotes d'interface Ethernet MTU et MRU dans la version 5.1.1 se comporte de la même manière pour les routeurs CRS et ASR 9000. Mais un routeur CRS qui exécute la version 5.1.1 n'inclut pas le CRC de 4 octets dans la valeur MTU et MRU affichée dans la sortie de **show controller**. Le comportement de la façon dont il est signalé n'est pas le même entre CRS et 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 dont le MTU et le MRU sont affichés dans la sortie show controller sur l'ASR 9000 changera à l'avenir de sorte que les 4 octets du CRC ne seront pas inclus dans la valeur MTU/MRU affichée. Ce changement futur peut être suivi avec l'ID de bogue Cisco [CSCuo93379](https://tools.cisco.com/bugsearch/bug/?bugid=93379).

Convertir la configuration lors de la mise à niveau d'une version antérieure à la version 5.1.1 vers la version 5.1.1 ou ultérieure

- MTU par défaut :

S'il y avait une interface principale sans sous-interface et sans commande `mtu` dans une version antérieure à la version 5.1.1 :

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

Et cette interface transporte des trames dot1q ou QinQ, alors le MTU doit être configuré manuellement à "mtu 1522" dans la version 5.1.1 et ultérieure :

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

Cette configuration permet de transporter les trames QinQ comme dans les versions précédentes. La valeur MTU peut être configurée sur 1518 si seuls dot1q et non QinQ doivent être transportés.

Si des sous-interfaces ont été configurées pour dot1q ou QinQ, mais avec le mot clé « any » et qu'aucune sous-interface QinQ avec 2 balises explicites n'a été configurée dans une version antérieure à 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 les versions 5.1.1 et ultérieures ne permettra de transporter que des trames avec une étiquette. Par conséquent, la MTU doit également être augmentée manuellement de 4 octets si des trames 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 QinQ avec 2 balises explicites (qui n'utilisent pas le mot clé « any ») est configurée, il n'est pas nécessaire de modifier la configuration MTU lors de la mise à niveau vers la version 5.1.1 et ultérieure :

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

S'il n'y a pas de sous-interface de transport de couche 2, mais uniquement des interfaces routées de couche 3, la configuration MTU devrait correspondre des deux côtés et il n'y aurait pas de trames plus grandes que la MTU transportée. Il n'est pas nécessaire de mettre à jour la configuration MTU lorsque vous effectuez une mise à niveau vers la version 5.1.1 et ultérieure.

- MTU non par défaut dans une version antérieure à la version 5.1.1 :

De même, lorsqu'une MTU non par défaut a été configurée dans une version antérieure à la version 5.1.1 et qu'aucune sous-interface n'a été configurée et que les trames dot1q ou QinQ doivent être transportées, la valeur MTU configurée doit être augmentée de 8 octets lors de la mise à niveau vers la version 5.1.1 ou ultérieure.

Version antérieure à la version 5.1.1 :

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

La MTU doit être augmentée manuellement de 8 octets lors de la mise à niveau vers la version 5.1.1 et ultérieure :

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

!

La valeur MTU configurée doit également être augmentée de 4 octets s'il existe une sous-interface dot1q et aucune sous-interface QinQ ou une sous-interface QinQ avec le mot clé any pour la balise second-dot1q.

Version antérieure à 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 ultérieure :

```
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 QinQ avec 2 balises explicites (qui n'utilisent pas le mot clé « any ») est configurée, il n'est pas nécessaire de modifier la configuration MTU lors de la mise à niveau vers la version 5.1.1 et ultérieure.

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

S'il n'y a pas de sous-interface de transport de couche 2, mais uniquement des interfaces routées de couche 3, la configuration MTU devrait correspondre des deux côtés et il n'y aurait pas de trames plus grandes que la MTU transportée. Il n'est pas nécessaire de mettre à jour la configuration MTU lorsque vous effectuez une mise à niveau vers la version 5.1.1 et ultérieure.

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.