

MTU-gedrag op Cisco IOS XR- en Cisco IOS-routers

Inhoud

[Inleiding](#)

[Achtergrondinformatie](#)

[Configureren](#)

[Vergelijking van Cisco IOS- en Cisco IOS XR-software](#)

[Routed L3-interfaces](#)

[Standaard MTU](#)

[Non-default MTU](#)

[Routed L3 sub-interfaces](#)

[L2VPN L2-interface](#)

[EVC \(ASR 9000\)](#)

[Non-EVC \(XR 12000 en CRS\)](#)

[Configuratie van automatische Ethernet-interfacedriver MTU en MRU](#)

[Converteer de configuratie wanneer u upgrade van een release eerder dan release 5.1.1 naar release 5.1.1 of later](#)

Inleiding

Dit document beschrijft het gedrag van de maximale transmissie-eenheid (MTU) op Cisco IOS[®] XR-routers en vergelijkt dat gedrag met Cisco IOS-routers. Het bespreekt ook MTU's op Routed Layer 3 (L3) interfaces en Layer 2 VPN (L2VPN) L2 interfaces die zowel de Ethernet Virtual Connection (EVC) en niet-EVC modellen gebruiken. Dit document beschrijft ook belangrijke veranderingen in hoe Ethernet interfacestuurprogramma MTU en maximum ontvangt eenheid (MRU) automatisch in release 5.1.1 en later worden geconfigureerd.

Achtergrondinformatie

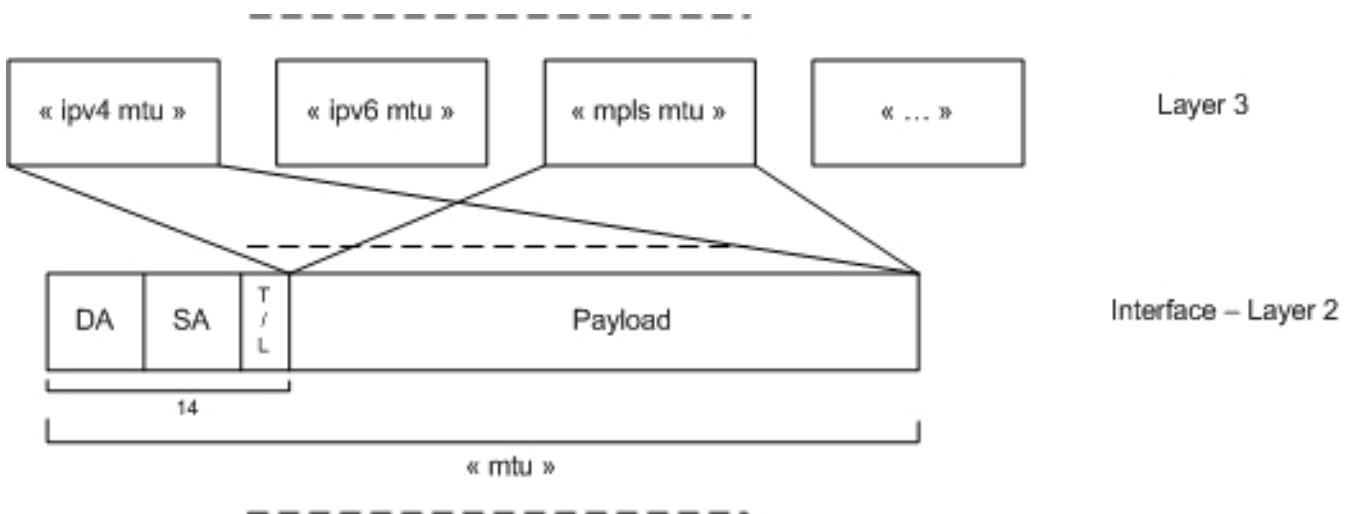
In een computernetwerk, bepaalt MTU van een communicatie protocol van een laag de grootte, in bytes, van de grootste eenheid van protocolgegevens die de laag over één interface mag overbrengen. Eén MTU-parameter is gekoppeld aan elke interface, laag en protocol.

De kenmerken van MTU in Cisco IOS XR-software zijn:

- MTU **configuratie** en **toon** opdrachten, bij L2 en L3, omvatten de headergrootte van hun corresponderende laag. De **mtu**-opdracht die de L2 MTU configureert, omvat bijvoorbeeld 14 bytes voor een Ethernet-interface (zonder dot1q), of 4 bytes voor Point-to-Point Protocol (PPP) of high-level datalink control (HDLC). De opdracht **ipv4 mtu** bevat 20 bytes van de

IPv4-header.

- De MTU van een hogere laag moet passen binnen de lading van de onderste laag. Bijvoorbeeld, als de interface MTU van een niet-dot1q Ethernet interface het gebrek van 1514 bytes is, dan kunnen de hogere laagprotocollen zoals Multiprotocol Label Switching (MPLS) een maximum MTU van 1500 bytes op die interface hebben. Dit betekent dat u slechts een 1500 bytes MPLS frame (inclusief labels) binnen het Ethernet frame kunt passen. U kunt geen 1508-byte MPLS MTU op die interface configureren als u twee MPLS-tags boven op een 1500-byte IPv4-pakket wilt toestaan. Om een 1508 bytes MPLS frame op een Ethernet interface te verzenden, moet de interface MTU worden verhoogd naar 1522, of hogere waarde, om ervoor te zorgen dat de L2 interface payload groot genoeg is om het MPLS frame te dragen.



- In de klassieke Cisco IOS-software (niet de Cisco IOS XR-software) vormt de opdracht **mtu** de L2-payloadgrootte, maar de L2-header is niet inbegrepen. Dit is anders dan de Cisco IOS XR-software die zowel de L2- als de L3-overhead in de opdracht **interface-mtu** bevat. De L3 MTU opdrachten, zoals in het geval van de **ipv4 mtu** opdracht, vormen de maximale pakketgrootte van dat protocol dat de L3 header bevat. Dit is vergelijkbaar met het geval van Cisco IOS XR-software.
- De standaardinterface MTU in de Cisco IOS XR-software moet het transport van een 1500 bytes L3-pakket toestaan. Daarom is de standaard MTU 1514 bytes voor een Ethernet-interface en 1504 bytes voor een seriële interface.

De rest van dit document illustreert MTU-kenmerken, vergelijkt Cisco IOS en Cisco IOS XR-softwaregedrag en geeft voorbeelden voor deze typen interfaces:

- Routed L3-interfaces
- Routed L3-subinterfaces
- L2VPN L2-interfaces

Configureren

Opmerking: Gebruik de [Command Lookup Tool](#) ([alleen geregistreerde](#) klanten) om meer informatie te verkrijgen over de opdrachten die in deze sectie worden gebruikt.

Opmerking: De [Output Interpreter Tool](#) ([alleen geregistreerde](#) klanten) ondersteunt bepaalde **show** opdrachten. Gebruik de Output Interpreter Tool om een analyse te bekijken van de output van de opdracht **show**.

Vergelijking van Cisco IOS- en Cisco IOS XR-software

In deze sectie worden Cisco IOS en Cisco IOS XR-softwaregedrag vergeleken met MTU-kenmerken.

In Cisco IOS-software bevatten de **mtu**-opdracht en de corresponderende **show**-opdrachten niet de L2-header. Gebruik de **mtu** opdracht om de L2 payload te configureren tot de maximale grootte voor de L3 pakketten, inclusief de L3 header.

Dit is niet hetzelfde als Cisco IOS XR-software, waarbij de **mtu**-opdracht de L2-header bevat (14 bytes voor Ethernet of 4 bytes voor PPP/HDLC).

Als een Cisco IOS-router met $mtu\ x$ is geconfigureerd *en met een Cisco IOS XR-router is verbonden, moet de corresponderende interface op de Cisco IOS XR-router worden geconfigureerd met $mtu\ x+14$ voor Ethernet-interfaces, of $mtu\ x+4$ voor seriële interfaces.*

Cisco IOS- en Cisco IOS XR-software hebben dezelfde betekenis voor de opdrachten **ipv4 mtu**, **ipv6 mtu** en **mpls mtu**; ze moeten met dezelfde waarden worden geconfigureerd.

Hierdoor wordt de configuratie in Cisco IOS-software op een Ethernet-interface weergegeven:

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

De corresponderende configuratie op de Cisco IOS XR-softwarebuur is:

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

Routed L3-interfaces

De MTU-waarden moeten hetzelfde zijn op alle apparaten die zijn aangesloten op een L2-netwerk. Anders kunnen deze symptomen worden gemeld:

- Intermediate System-to-Intermediate System (IS-IS) nabijheid komen niet naar boven. Standaard gebruikt IS-IS hello-padding; daarom kunnen hellos gekarakteriseerd worden als giganten en kunnen ze worden weggelaten als een router een MTU-waarde heeft die lager is dan de waarden bij de andere routers.
- Open Shortest Path First (OSPF)-nabijheid komen vast te zitten in de staat Exstart of Exchange, omdat grote DBD-pakketten (Database Description) kunnen worden gekarakteriseerd als giganten en kunnen worden gedropt. Wanneer de pakketten worden ontvangen op een router met een lagere MTU waarde, zijn de databases niet gesynchroniseerd.

- Het gegevensverkeer wordt gekarakteriseerd als reuzen en gevallen wanneer het op een apparaat met een waarde wordt ontvangen MTU die lager is dan bij het overbrengende apparaat.
- Er is een lage doorvoersnelheid wanneer grote pakketten worden gedropt. In het geval van pad MTU ontdekking, de TCP sessie kan herstellen wanneer grote pakketten worden gedropt, maar dit beïnvloedt de doorvoersnelheid.

Standaard MTU

Deze paragraaf analyseert de standaard MTU van een gerouteerde interface wanneer de `mtu` opdracht niet geconfigureerd is:

```
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#
```

In dit voorbeeld, de standaard L2 interface MTU is 1514 bytes en omvat 14 bytes van Ethernet kopbal. De 14 bytes worden gerekend door 6 bytes van het adres van bestemmingsMAC, 6 bytes van het adres van bronMAC, en 2 bytes van type of lengte. Dit omvat niet de inleiding, de kaderafbakening, 4 bytes van de opeenvolging van de kadercontrole (FCS), en interframe hiaat. Voor een PPP- of HDLC-frame worden 4 bytes van de L2-header verwerkt; de standaardinterface MTU is dus 1504 bytes.

De L3 kindprotocollen erven hun MTU van de lading van de ouder MTU. Als je 14 bytes van een L2 header aftrekt van een L2 MTU van 1514 bytes, heb je een L2 payload van 1500 bytes. Dit wordt de MTU voor L3-protocollen. IPv4, IPv6, MPLS en Connectionless Network Service (CLNS) erven deze 1500 bytes MTU. Hierdoor kan een Cisco IOS XR Ethernet-interface standaard een 1500 byte L3-pakket transporteren dat hetzelfde is als de standaard applicatie op een Cisco IOS Ethernet-interface.

Non-default MTU

Deze paragraaf laat zien hoe u een **mpls mtu** van 1508 kunt configureren om een IPv4-pakket van 1500 bytes te verzenden met twee MPLS-tags van 4 bytes elk, bovenop het pakket:

```
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#
```

Hoewel de opdracht **mpls mtu 1508** is vastgelegd, wordt deze niet toegepast, omdat MPLS nog steeds een MTU van 1500 bytes heeft in de opdracht **show**. Dit komt doordat L3-kinderprotocollen geen MTU kunnen hebben die groter is dan de payload van hun parent L2-interface.

Om twee labels boven op een IP-pakket van 1500 bytes toe te staan, moet u:

- Configureer een L2 interface MTU van 1522 bytes, zodat alle onderliggende protocollen (inclusief MPLS) een MTU van 1508 bytes erven ($1522 - 14 = 1508$).
- Verminder de MTU van de L3 protocollen tot 1500 bytes, zodat alleen MPLS 1500 bytes mag overschrijden.

```
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#
```

Met deze configuratie kunt u IPv4- en IPv6-pakketten van 1500 bytes en MPLS-pakketten van 1508 bytes (een pakket van 1500 bytes met twee tags aan de bovenkant) verzenden.

Routed L3 sub-interfaces

Deze kenmerken zijn van toepassing op gerouteerde L3 subinterfaces.

Een gerouteerde sub-interface MTU erft MTU van zijn ouder belangrijkste interface; voeg 4 bytes toe voor elke VLAN-tag die op de sub-interface is geconfigureerd. Er zijn dus 4 bytes voor een dot1q subinterface en 8 bytes voor een IEEE 802.1Q-subinterface (QinQ).

Hierdoor kunnen L3-pakketten van dezelfde grootte zowel op de hoofdinterface als op de subinterface worden doorgestuurd.

Het **mtu** commando kan geconfigureerd worden onder de subinterface, maar het wordt alleen toegepast als het lager is of gelijk is aan de MTU die geërfd is van de hoofdinterface.

Dit is een voorbeeld waar MTU van de belangrijkste interface 2000 bytes is:

```
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#
```

In de **show** commando's is de MTU van de subinterface 2004; voeg 4 bytes toe aan de MTU van

de hoofdinterface omdat er één dot1q tag is geconfigureerd onder de subinterface.

Echter, de MTU van de IPv4 en IPv6 pakketten zijn nog steeds hetzelfde als die van de hoofdinterface (1986). Dit komt doordat de MTU van de L3 protocollen nu wordt berekend als: $2004 - 14 - 4 = 1986$.

De **mtu** opdracht kan worden geconfigureerd onder de subinterface, maar de geconfigureerde MTU wordt alleen toegepast als deze lager is of gelijk is aan de MTU die wordt geërfd van de hoofdinterface (4 bytes groter dan de MTU van de hoofdinterface).

Wanneer de MTU van de subinterface die groter is dan de geërfde MTU, wordt deze niet toegepast, zoals hier wordt getoond:

```
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#
```

Aldus, kunt u slechts het **mtu** bevel gebruiken om de waarde te verminderen MTU die van de belangrijkste interface wordt geërfd.

Op dezelfde manier kunt u ook de MTU-opdrachten van L3-protocollen (IPv4, IPv6, MPLS) gebruiken om de waarde van de L3 MTU te verlagen die van de subinterface L2-payload wordt geërfd. Het L3 protocol MTU wordt niet van kracht wanneer het is ingesteld op een waarde die niet past in de payload van de L2 MTU.

L2VPN L2-interface

De MTU voor een L2VPN is belangrijk omdat Label Distribution Protocol (LDP) geen pseudowire (PW) omhoog brengt wanneer de MTU's op de bevestigingscircuits aan elke kant van een PW niet hetzelfde zijn.

Hier is een **show** commando dat illustreert dat een L2VPN PW beneden blijft wanneer er een MTU mismatch is:

```
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#
```

In dit voorbeeld, merk op dat de MPLS L2VPN provider randen (PEs) aan elke kant de zelfde MTU waarde moet signaleren om PW omhoog te brengen.

De MTU zoals aangegeven door MPLS LDP bevat geen L2-overhead. Dit is anders dan de XR interface **config** en **toont** opdrachten die de L2 overhead bevatten. De MTU op de subinterface is 2018 bytes (zoals geërfd van de hoofdinterface van 2014 bytes), maar LDP signaleerde een MTU van 2000 bytes. Hierdoor worden 18 bytes (14 bytes van Ethernet-header + 4 bytes van 1 dot1q-tag) afgetrokken van de L2-header.

Het is belangrijk om te begrijpen hoe elk apparaat de waarden MTU van de gehechtheidscircuits berekent om MTU mismatches te bevestigen. Dit is afhankelijk van parameters zoals leverancier, platform, softwareversie en configuratie.

EVC (ASR 9000)

De Cisco ASR 9000 Series aggregatieservices router gebruikt het EVC-infrastructuurmodel, waarmee flexibel VLAN kan worden afgestemd op L2VPN L2-interfaces en subinterfaces.

De EVC L2VPN L2 interfaces hebben deze kenmerken:

- Zij staan configuratie van één of meerdere markeringen met het bevel van de **inkapseling toe**.
- Door gebrek en met enkel het bevel van de **inkapseling**, worden de markeringen bewaard en over PWs vervoerd. Hierdoor hoeft u de tags niet standaard te strippen, zoals u dat op niet-EVC-platforms moet doen.
- Gebruik de opdracht **Herschrijven** wanneer u besluit de inkomende tags te pop of om een aantal extra tags boven op het inkomende frame te drukken.

Om de subinterface MTU te berekenen, neem de belangrijkste interface MTU (of het gebrek of manueel gevormd onder de belangrijkste interface), en voeg 4 bytes voor elke die markering van VLAN toe met het bevel van de **inkapseling** wordt gevormd. Zie [Specifieke EFP-insluitingsopdrachten](#).

Wanneer er een **mtu**-opdracht onder de subinterface is, wordt deze alleen van kracht als deze lager is dan de berekende MTU. De opdracht **herschrijven** heeft geen invloed op de MTU van de subinterface.

Hierna volgt een voorbeeld:

```
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)
```

In dit voorbeeld is de MTU in de hoofdinterface 2014 bytes; voeg 8 bytes toe omdat er twee tags zijn geconfigureerd onder de subinterface.

Als u **mtu 2026** bytes configureert onder de subinterface, wordt deze niet toegepast omdat deze groter is dan de MTU van de subinterface die is geërfd van de hoofdinterface (2022). Hierdoor kunt u alleen een subinterface MTU configureren die lager is dan 2022 bytes.

Gebaseerd op deze sub-interface MTU, bereken MTU van de MPLS LDP payload die aan de buur wordt gesignaleerd, en zorg ervoor dat het identiek is aan die berekend door de externe L2VPN PE. Dit is waar het **herschrijf** bevel in spel komt.

Om de MTU van de MPLS LDP-payload te berekenen, neemt u de MTU van de subinterface en:

1. Trek 14 bytes af voor de Ethernet-header.
2. Trek 4 bytes af voor elke tag die in de opdracht **Herschrijven is** opgenomen en die onder de subinterface is geconfigureerd.
3. Voeg 4 bytes toe voor elke tag die in de opdracht **herschrijven is** gedrukt en die onder de subinterface is geconfigureerd.

Dit is hetzelfde voorbeeld met de QinQ-configuratie op 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)
```

Dit is de berekening voor de MTU van de MPLS LDP payload:

1. MTU-waarde van de subinterface MTU: 2022 bytes
2. Aftrekken 14 bytes Ethernet-header: $2022 - 14 = 2008$ bytes
3. Trek 4 bytes af voor elke gekopieerde tag in de **herschrijven** commando: $2008 - 4 * 2 = 2000$

Zorg ervoor dat de afstandsbediening een MPLS LDP-payload van 2000 bytes adverteert. Anders past u de MTU-grootte van het lokale of externe bijlagecircuit (AC) aan, zodat deze overeenkomt.

```
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
```

Opdrachten voor specifieke Ethernet Flow Point (EFP)-insluiting

Deze insluitingen tellen als het aanpassen nul markeringen, zodat verhogen zij niet de sub-interface MTU:

- inkapseling ongetagd
- inkapselingsstandaard

Deze inkapselingsbepalingen hebben geen invloed op het aantal tags dat nodig is om de subinterface MTU te berekenen:

- inboorlinge

- met payload-ethertype
- exact
- kosten
- toegang bron-mac of toegang bestemming-mac

inkapseling [dot1q|dot1ad] met prioriteitslabels als overeenkomend met één tag.

Het 'enige' sleutelwoord dat gebruikt wordt als de innermost tag match verhoogt de subinterface MTU niet.

- **inkapseling dot1q om het even welk** verhoogt niet sub-interface MTU.
- **inkapseling dot1ad 10 dot1q elk** als één tag wordt meegeteld; het verhoogt de subinterface MTU met 4 bytes.
- **inkapseling dot1ad elke dot1q 7** wordt als twee tags geregistreerd; het verhoogt de sub-interface MTU met 8 bytes.

De bereiken van VLAN-ID's verhogen de subinterface MTU:

- **inkapseling dot1q 10-100** wordt als één tag geteld; het verhoogt de subinterface MTU met 4 bytes.

De inkapseling MTU overhead van een EFP die een disjunctive match is wordt behandeld als de MTU van zijn hoogste element.

- **inkapseling dot1q 10-100, untagged** is als één tag omdat de bereik 10-100 het hoogste element is.

Non-EVC (XR 12000 en CRS)

Routers zoals Cisco XR 12000 Series router en Carrier Routing System (CRS) gebruiken de traditionele configuratie voor VLAN-matching op subinterfaces. Deze kenmerken zijn van toepassing op L2VPN L2-interfaces op CRS en op XR 12000-routers die niet volgens het EVC-model werken:

- Op niet-EVC-platforms worden de inkomende dot1q- of dot1ad-tags automatisch gestript wanneer ze worden ontvangen op een L2-transportsubinterface.
- Wanneer u de payload-grootte berekent voor MPLS LDP om te signaleren, trekt u de grootte van de tags af van de MTU van de subinterface, zoals in de opdracht **show** interface wordt getoond.
- Dit is vergelijkbaar met het geval van een gerouteerde subinterface.
- De subinterface erft zijn MTU van de hoofdinterface; voeg de 4 bytes voor elke tag toe aan de MTU van de hoofdinterface om de MTU van de subinterface te berekenen. Als een QinQ subinterface bijvoorbeeld 2 dot1q tags heeft, heeft de subinterface standaard een MTU die 8 bytes groter is dan de MTU van de hoofdinterface.
- U kunt ook de **mtu**-opdracht gebruiken onder de subinterface, maar het wordt alleen gebruikt om de MTU van de subinterface te verminderen, die wordt geërfd van de MTU van de hoofdinterface.

Hier zijn een aantal voorbeelden die deze kenmerken illustreren.

Dit voorbeeld laat zien hoe een niet-EVC subinterface is geconfigureerd:

```
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#
```

De niet-EVC-platforms gebruiken de opdrachten **dot1q VLAN** of **dot1ad VLAN** in plaats van de opdrachten **inkapseling** en **herschrijven** van de EVC-platforms (ASR 9000).

Als u een MTU niet expliciet configureert op de hoofd- of subinterface, kan een 1500 bytes L3-pakket standaard worden ontvangen:

```
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#
```

De subinterface MTU wordt berekend vanuit de hoofdinterface MTU (1514); voeg 4 bytes toe voor elke dot1q tag. Omdat er één tag op de subinterface is geconfigureerd met de opdracht **dot1q VLAN 201**, voeg 4 bytes toe aan 1514 voor een MTU van 1518 bytes.

De bijbehorende payload MTU in MPLS LDP is 1500 bytes, omdat de 14 bytes van Ethernet-header niet worden geteld en de één dot1q-tag automatisch wordt gekopieerd door het niet-EVC-platform wanneer deze over de PW gaat:

```
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
```

Als u de MTU van de hoofdinterface verhoogt naar 2014 bytes, wordt de MTU van de subinterface dienovereenkomstig verhoogd:

```
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
```

Dus om de MPLS LDP MTU te berekenen, trekt u 14 bytes Ethernet-header af en voegt u 4 bytes toe voor elke tag die onder de subinterface is geconfigureerd.

Configuratie van automatische Ethernet-interfacedriver MTU en MRU

Op Ethernet interfaces is de interface driver geconfigureerd met een MTU en MRU die gebaseerd is op de interface MTU configuratie.

De geconfigureerde MTU en MRU op het Ethernet-interfacestuurprogramma kunnen worden gezien met de **showcontroller <interface> all** commando.

In releases eerder dan Cisco IOS XR release 5.1.1 zijn de MTU en MRU op het Ethernet-interfacestuurprogramma automatisch geconfigureerd op basis van de Cisco IOS XR MTU-configuratie op de interface.

De MTU/MRU geconfigureerd op de Ethernet driver was gewoon gebaseerd op de geconfigureerde MTU + 12 bytes voor de toevoeging van 2 Ethernet-tags en het CRC-veld. De 12 bytes werden toegevoegd aan de Ethernet driver MTU/MRU, ongeacht of er VLAN-tags waren geconfigureerd op de subinterfaces.

Hier wordt een voorbeeld getoond met alle Cisco IOS XR-versies eerder dan Cisco IOS XR release 5.1.1 en een standaard-MTU van 1514 op een ASR 9000 interface:

```
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>
```

In Cisco IOS XR-release 5.1.1 en hoger zijn de MTU en MRU die op het Ethernet-interfacestuurprogramma worden gebruikt, gewijzigd en nu gebaseerd op het aantal VLAN-tags

die op een van de subinterfaces zijn geconfigureerd.

Als er geen VLAN-tags op een subinterface zijn geconfigureerd, is de driver MTU/MRU gelijk aan de geconfigureerde MTU op interface + 4 CRC-bytes, bijvoorbeeld $1514 + 4 = 1518$ bytes.

Als één VLAN op een subinterface is geconfigureerd, is de driver MTU/MRU gelijk aan de geconfigureerde MTU + 8 bytes (1 tag + CRC), bijvoorbeeld $1514 + 8 = 1522$ bytes.

Als twee VLAN-tags op een subinterface zijn geconfigureerd, is de driver MTU/MRU gelijk aan de geconfigureerde MTU + 12 bytes (2 tags + CRC), bijvoorbeeld $1514 + 12 = 1526$ bytes

Als QinQ met het **willekeurige** sleutelwoord is geconfigureerd voor de tweede-dot1q-tag, is het stuurprogramma MTU/MRU gelijk aan de geconfigureerde MTU + 8 bytes (1 tag + CRC), bijvoorbeeld $1514 + 8 = 1522$ bytes.

Deze voorbeelden tonen het gedrag in Cisco IOS XR release 5.1.1 en hoger op een 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>
```

In de meeste situaties zou dit gedrag in release 5.1.1 en later geen veranderingen in de configuratie van de MTU op de interface moeten vereisen.

Deze gedragsverandering kan problemen in het geval van een sub-interface veroorzaken die met één enkele markering van VLAN wordt gevormd, maar ontvangt pakketten met twee markeringen van VLAN. In die situatie kunnen de ontvangen pakketten de MRU op de Ethernet interfacestuurprogramma overschrijden. Om die voorwaarde te elimineren, kan interface MTU of met 4 bytes worden verhoogd of de sub-interface die met twee markeringen van VLAN wordt gevormd.

De automatische configuratie van Ethernet-interfacestuurprogramma's MTU en MRU in het gedrag van release 5.1.1 is hetzelfde voor een CRS- en ASR 9000-router. Maar een CRS router die release 5.1.1 in werking stelt omvat niet de 4-byte CRC in de MTU en MRU waarde die in de output van de **showcontroller** wordt weergegeven. Het gedrag van hoe het wordt gerapporteerd is niet hetzelfde tussen CRS en 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)
```

De manier waarop de MTU en MRU worden weergegeven in de output van de showcontroller op de ASR 9000 zal in de toekomst veranderen zodat de 4 bytes van CRC niet zullen worden opgenomen in de weergegeven MTU/MRU waarde. Deze toekomstige wijziging kan worden gevolgd met Cisco bug-id [CSCuo93379](#).

Converteer de configuratie wanneer u upgrade van een release eerder dan release 5.1.1 naar release 5.1.1 of later

- Standaard MTU:

Als er een hoofdinterface was zonder subinterface en zonder **mtu**-opdracht in een release eerder dan release 5.1.1:

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

En deze interface transporteert dot1q of QinQ frames, dan zou MTU handmatig moeten worden geconfigureerd om "mtu 1522" in release 5.1.1 en hoger:

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

Deze configuratie maakt het mogelijk om QinQ-frames te transporteren zoals in de eerdere releases. De MTU-waarde kan worden ingesteld op 1518 als alleen dot1q en niet QinQ worden getransporteerd.

Als er subinterfaces geconfigureerd waren voor dot1q of QinQ, maar met het "enige" trefwoord en er geen QinQ subinterfaces met 2 expliciete tags geconfigureerd waren in een release eerder dan release 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
```

!

Deze configuratie in release 5.1.1 en hoger zal alleen toestaan om frames met één tag te transporteren, zodat de MTU ook handmatig met 4 bytes moet worden verhoogd als QinQ-frames moeten worden getransporteerd:

```
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
!
```

Als een QinQ subinterface met 2 expliciete tags (die niet het "enige" trefwoord gebruiken) is geconfigureerd, hoeft de MTU-configuratie niet te worden aangepast wanneer u upgrade naar release 5.1.1 en hoger:

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

Als er geen L2 transport subinterface maar alleen L3 routed interfaces is, wordt verwacht dat de MTU configuratie aan beide kanten zou overeenkomen en er zouden geen frames groter zijn dan de MTU die wordt getransporteerd. Het is niet nodig om de MTU-configuratie bij te werken wanneer u upgrade naar release 5.1.1 en hoger.

- Non-default MTU in release eerder dan release 5.1.1:

Op dezelfde manier wanneer een niet-standaard MTU werd geconfigureerd in een release eerder dan release 5.1.1 en er geen subinterface is geconfigureerd en dot1q of QinQ frames moeten worden getransporteerd, dan moet de geconfigureerde MTU-waarde worden verhoogd met 8 bytes wanneer u upgrade naar release 5.1.1 of hoger.

Release eerder dan release 5.1.1:

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

De MTU moet handmatig met 8 bytes worden verhoogd wanneer u upgrade naar release 5.1.1 en hoger:

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

De ingestelde MTU-waarde moet ook worden verhoogd met 4 bytes als er een dot1q subinterface en geen QinQ subinterface of een QinQ subinterface is met het willekeurige trefwoord voor de

second-dot1q tag.

Release eerder dan release 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
!
```

Release 5.1.1 en hoger:

```
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
!
```

Als een QinQ sub-interface met 2 expliciete markeringen (die niet het "enige" sleutelwoord gebruiken) wordt gevormd, is er geen behoefte om de configuratie te wijzigen MTU wanneer u aan release 5.1.1 en later verbetert.

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

Als er geen L2 transport subinterface is, maar alleen L3 routed interfaces, wordt verwacht dat de MTU configuratie aan beide kanten zou overeenkomen en er zouden geen frames groter zijn dan de MTU die wordt getransporteerd. Het is niet nodig om de MTU-configuratie bij te werken wanneer u upgrade naar release 5.1.1 en hoger.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.