

Поведение MTU на Cisco IOS XR и маршрутизаторах Cisco IOS

Содержание

[Введение](#)

[Общие сведения](#)

[Настройка](#)

[Сравнение Cisco IOS и программного обеспечения Cisco IOS XR](#)

[Направленные L3 интерфейсы](#)

[MTU по умолчанию](#)

[MTU на по умолчанию](#)

[Направленные подчиненные интерфейсы L3](#)

[Интерфейс L2VPN L2](#)

[EVC \(ASR9000\)](#)

[Non EVC \(XR 12000 и CRS\)](#)

[Автоматический MTU драйвера интерфейса Ethernet и конфигурация MRU](#)

[Преобразуйте конфигурацию, когда вы обновите от выпуска ранее, чем выпуск 5.1.1 к выпуску 5.1.1 или позже](#)

Введение

Этот документ описывает способы поведения максимального размера передаваемого блока данных (MTU) на Cisco маршрутизаторы ^{IOS® XR} и сравнивает те способы поведения с маршрутизаторами Cisco IOS. Это также обсуждает MTU на маршрутизированных интерфейсах 3 уровня (L3) и Сети VPN второго уровня (L2VPN) интерфейсы L2, которые используют и Виртуальное соединение Ethernet (EVC) и non-EVC модели. Этот документ также описывает важные изменения к тому, как MTU Драйвера интерфейса Ethernet и Maximum Receive Unit (MRU) автоматически настроены в Выпуске 5.1.1 и позже.

Общие сведения

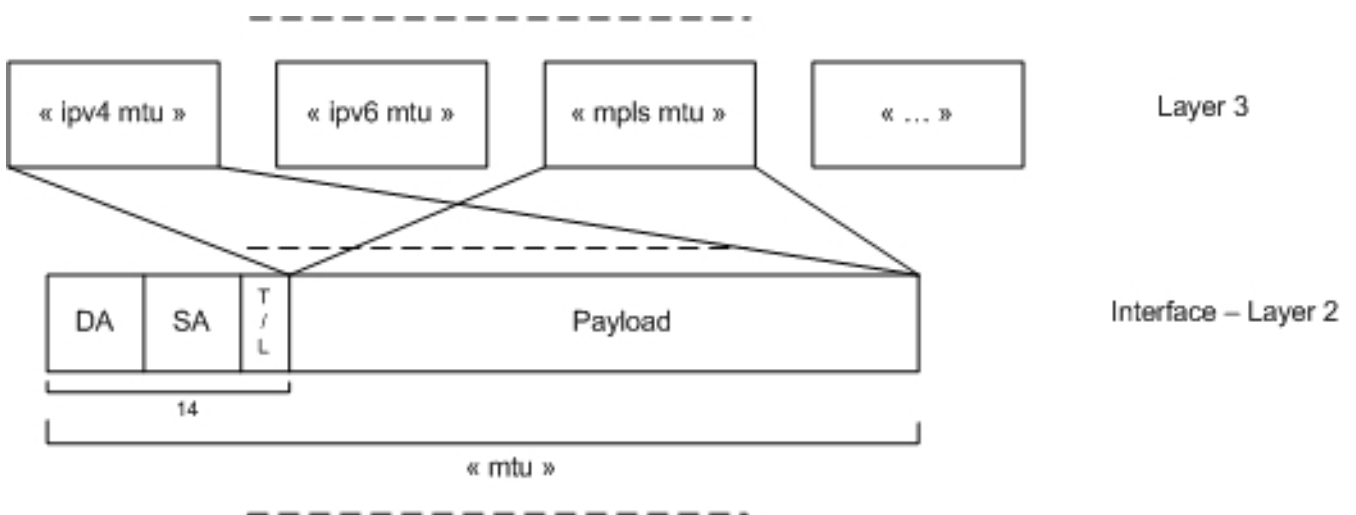
В компьютерных сетях MTU протокола связи уровня определяет размер, в байтах, самой большой протокольной информационной единицы, которую уровню позволяют передать по одному интерфейсу. Один параметр MTU привязан к каждому интерфейсу, уровню и протоколу.

Характеристики MTU в программном обеспечении Cisco IOS XR:

- **Config MTU и команды показа**, в L2 и L3, включают размер заголовка их соответствующего уровня. Например, команда `mtu`, которая настраивает L2 MTU,

включает 14 байтов для Интерфейса Ethernet (без dot1q) или 4 байта для Протокола PPP или высокоуровневого протокола управления каналом передачи данных (HDLC).
Команда `ipv4 mtu` включает 20 байтов заголовка IPv4.

- MTU более высокого уровня должен соответствовать в информационном наполнении низшего уровня. Например, если максимальный размер блока данных (MTU) интерфейса Интерфейса Ethernet не-dot1q является по умолчанию 1514 байтов, то протоколы высшего уровня, такие как Многопротокольная коммутация по меткам (MPLS) могут иметь максимальное значение размера блока данных 1500 байтов на том интерфейсе. Это означает, что можно соответствовать только 1500-байтовому кадру MPLS (включая метки) во Фрейме Ethernet. Если вы хотите позволить две метки MPLS поверх 1500-байтового Пакета IPV4, вы не можете настроить 1508-байтового MPLS MTU на том интерфейсе. Для передачи 1508-байтового кадра MPLS на Интерфейсе Ethernet максимальный размер блока данных (MTU) интерфейса должен быть увеличен до 1522 или более высокого значения, чтобы гарантировать, что информационное наполнение интерфейса L2 является достаточно большим для переноса кадра MPLS.



- В классическом программном обеспечении Cisco IOS (не программное обеспечение Cisco IOS XR), интерфейсная **команда `mtu`** настраивает объем полезных данных L2, но не включает заголовок L2. Это отличается от программного обеспечения Cisco IOS XR, которое включает и L2 и издержки L3 в **интерфейсной команде `mtu`**. Команды L3 MTU, как в случае **команды `ipv4 mtu`**, настраивают максимальный размер пакета того протокола, который включает заголовок L3. Это подобно случаю программного обеспечения Cisco IOS XR.
- MTU интерфейса по умолчанию в программном обеспечении Cisco IOS XR должен позволить транспорт 1500-байтового пакета L3. Поэтому MTU по умолчанию составляет 1514 байтов для основного Интерфейса Ethernet и 1504 байта для последовательного интерфейса.

Оставшаяся часть этого документа иллюстрирует характеристики MTU, сравнивает Cisco IOS и поведение программного обеспечения Cisco IOS XR, и дает примеры для этих типов интерфейсов:

- Направленные L3 интерфейсы
- Направленные подчиненные интерфейсы L3

- Интерфейсы L2VPN L2

Настройка

Примечание: [Чтобы получить подробные сведения о командах в данном документе, используйте Средство поиска команд \(только для зарегистрированных клиентов\).](#)

Примечание: [Средство интерпретации выходных данных \(только зарегистрированные клиенты\)](#) поддерживает некоторые команды `show`. Используйте Средство интерпретации выходных данных, чтобы просмотреть анализ выходных данных команды `show`.

Сравнение Cisco IOS и программного обеспечения Cisco IOS XR

Этот раздел сравнивает Cisco IOS и поведение программного обеспечения Cisco IOS XR в отношении характеристик MTU.

В программном обеспечении Cisco IOS команда `mtu` и соответствующие команды `показа` не включают заголовок L2. Используйте команду `mtu` для настройки информационного наполнения L2 к максимальному размеру для пакетов L3, включая заголовок L3.

Это отличается от программного обеспечения Cisco IOS XR, где команда `mtu` включает заголовок L2 (14 байтов для Ethernet или 4 байта для PPP/HDLC).

Если маршрутизатор Cisco IOS настроен с `mtu x` и связан с маршрутизатором Cisco IOS XR, то соответствующий интерфейс на маршрутизаторе Cisco IOS XR должен быть настроен с `mtu x+14` для Интерфейсов Ethernet или `mtu x+4` для последовательных интерфейсов.

Cisco IOS и программное обеспечение Cisco IOS XR имеют то же значение для команд `ipv4 mtu`, `ipv6 mtu` and `mpls mtu`; они должны быть настроены с теми же значениями.

В результате это - конфигурация в программном обеспечении Cisco IOS на Интерфейсе Ethernet:

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

Соответствующая конфигурация на соседнем узле программного обеспечения Cisco IOS XR:

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

Направленные L3 интерфейсы

Значения MTU должны быть тем же на всех устройствах, связанных с сетью L2. В противном случае об этих признаках можно было бы сообщить:

- Смежности Обмена информацией между промежуточными системами (IS-IS) не подходят. По умолчанию IS-IS использует hello-padding; поэтому, hellos мог бы быть охарактеризован как Giant и мог бы быть отброшен, когда один маршрутизатор имеет значение MTU, которое ниже, чем значения в других маршрутизаторах.
- Смежности Протокола OSPF застревают в Exstart или Состоянии обмена, потому что большой дескриптор базы данных (DBD) пакеты могли бы быть охарактеризованы как Giant и могли бы быть отброшены. Когда пакеты получены на маршрутизаторе со значением более низкого MTU, базы данных "not synchronized".
- Трафик данных характеризуется как Giant и отброшен, когда он получен на устройстве со значением MTU, которое ниже, чем тот в устройстве передачи.
- Когда большие пакеты отброшены, существует низкая пропускная способность. В случае обнаружения MTU-маршрута может восстановиться сеанс TCP, когда большие пакеты отброшены, но это влияет на пропускную способность.

MTU по умолчанию

Когда команда `mtu` не настроена, этот раздел анализирует MTU по умолчанию маршрутизируемого интерфейса:

```
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#

```

В данном примере максимальный размер блока данных (MTU) интерфейса L2 по умолчанию составляет 1514 байтов и включает 14 байтов Заголовка ethernet. 14 байтов считаются на 6 байтов MAC - адреса назначения, 6 байтов источника с MAC-адресом и 2 байта типа или длины. Это не включает преамбулу, разграничителя кадров, 4 байта контрольной суммы фрейма (FCS) и межкадровый разрыв. Для кадра PPP или кадра HDLC, составляются 4 байта заголовка L2; таким образом, MTU интерфейса по умолчанию составляет 1504 байта.

Дочерние протоколы L3 наследовали свой MTU от информационного наполнения родительского MTU. При вычитании 14 байтов заголовка L2 от L2 MTU 1514 байтов у вас есть информационное наполнение L2 1500 байтов. Это становится MTU для протоколов L3. IPv4, IPv6, MPLS и Обслуживание сети без установления соединения (CLNS) наследовали this1500 MTU байта. В результате Интерфейс Ethernet Cisco IOS XR, по умолчанию, может транспортировать 1500-байтовый пакет L3, который совпадает с default на Интерфейсе Ethernet Cisco IOS.

MTU ня по умолчанию

Этот раздел показывает, как настроить **mpls mtu 1508** для передачи Пакета IPV4 1500 байтов с двумя метками MPLS 4 байтов каждый поверх пакета:

```

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#
```

Несмотря на то, что команда **mpls mtu 1508** была передана, она не применена, потому что MPLS все еще имеет MTU 1500 байтов в команде **показа**. Это вызвано тем, что дочерние протоколы L3 не могут иметь MTU, больше, чем информационное наполнение их родительского интерфейса L2.

Для разрешения двух меток поверх пакета байтового IP 1500 байта вы должны:

- Настройте максимальный размер блока данных (MTU) интерфейса L2 1522 байтов, так, чтобы все дочерние протоколы (включая MPLS) наследовали MTU 1508 байтов (1522 - 14 = 1508).
- Уменьшите MTU протоколов L3 к 1500 байтам, так, чтобы только MPLS позволили превысить 1500 байтов.

```
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#
```

Эта конфигурация позволяет вам передать IPv4 и пакеты IPv6 1500 байтов, и пакеты MPLS 1508 байтов (пакет в 1 байт 1500 года с два наклеивает вершину).

Направленные подчиненные интерфейсы L3

Эти характеристики применяются к маршрутизированным подчиненным интерфейсам L3.

Маршрутизированный MTU подчиненного интерфейса наследовал MTU своего родительского основного интерфейса; добавьте 4 байта для каждого тега VLAN, настроенного на подчиненном интерфейсе. Таким образом существует 4 байта для подчиненного интерфейса dot1q и 8 байтов для IEEE 802.1Q, туннелирующего (QinQ) подчиненный интерфейс.

В результате пакеты L3 одинакового размера могут быть переданы и на основном интерфейсе и на подчиненном интерфейсе.

Команда mtu может быть настроена под подчиненным интерфейсом, но это применено, только если это ниже или равно MTU, который наследован от основного интерфейса.

Это - пример, где MTU основного интерфейса составляет 2000 байтов:

```
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#
```

В командах показа MTU подчиненного интерфейса является 2004; добавьте 4 байта к MTU основного интерфейса, потому что существует одна метка dot1q, настроенная под подчиненным интерфейсом.

Однако MTU IPv4 и пакетов IPv6 является все еще тем же как тем из основного интерфейса (1986). Это вызвано тем, что MTU протоколов L3 теперь вычислен как: $2004 - 14 - 4 = 1986$.

Команда mtu может быть настроена под подчиненным интерфейсом, но настроенный MTU применен, только если это ниже или равно MTU, который наследован от основного интерфейса (на 4 байта больше, чем MTU основного интерфейса).

Когда MTU подчиненного интерфейса, который больше, чем наследованный MTU, он не применен, как показано здесь:

```
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#
```

Таким образом можно использовать только **команду mtu** для понижения значения MTU, наследованного от основного интерфейса.

Точно так же можно также использовать команды MTU протоколов L3 (IPv4, IPv6, MPLS) для понижения значения MTU L3, наследованного от подчиненного интерфейса информационное наполнение L2. MTU протокола L3 не вступает в силу, когда он настроен к значению, которое не помещается в информационное наполнение L2 MTU.

Интерфейс L2VPN L2

MTU для L2VPN важен, потому что Протокол распределения меток (LDP) не приносит pseudowire (PW), когда MTU на прикрепляемых каналах в каждой стороне PW не являются тем же.

Вот **команда показа**, которая иллюстрирует, что L2VPN, PW остается на второй год, когда существует несоответствие 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#
```

В данном примере обратите внимание, что границы провайдера L2VPN MPLS (PE) в каждой стороне должны сигнализировать то же значение MTU для внедрения PW.

MTU, сообщенный LDP MPLS, не включает издержки L2. Это отличается от **config** интерфейса XR и **команд показа**, которые включают издержки L2. MTU на подчиненном интерфейсе составляет 2018 байтов (как наследовано от основного интерфейса 2014 байтов), но LDP сигнализировал MTU 2000 байтов. В результате это вычитает 18 байтов (14 байтов Заголовка ethernet + 4 байта из 1 метки dot1q) от заголовка L2.

Важно понять, как каждое устройство вычисляет значения MTU прикрепляемых каналов для решения проблемы несоответствий MTU. Это зависит от параметров, таких как поставщик, платформа, версия программного обеспечения и конфигурация.

EVC (ASR9000)

Маршрутизатор агрегации Cisco ASR серии 9000 использует модель инфраструктуры EVC, которая позволяет гибкую VLAN, совпадающую на интерфейсах L2VPN L2 и подчиненных интерфейсах.

Интерфейсы EVC L2VPN L2 имеют эти характеристики:

- Они позволяют конфигурацию одной или более меток с **командой encapsulation**.
- По умолчанию и только с **командой encapsulation**, метки сохранены и транспортированы по PWs. В результате вы не должны разделять метки по умолчанию, как необходимо сделать на non-EVC платформах.
- Используйте команду **перезаписи**, когда вы решите вытолкнуть входящие метки или выдвинуть некоторые дополнительные метки поверх входящего фрейма.

Для вычислений MTU подчиненного интерфейса возьмите MTU основного интерфейса (или по умолчанию или тот, настроенный вручную под основным интерфейсом), и добавьте 4 байта для каждого тега VLAN, настроенного с **командой encapsulation**. Посмотрите [Определенные Команды Encapsulation EFP](#).

Когда существует **команда mtu** под подчиненным интерфейсом, она вступает в силу, только если это ниже, чем вычисленный MTU. Команда **перезаписи** не влияет на MTU подчиненного интерфейса.

Например:

```
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)
```

В данном примере MTU в основном интерфейсе составляет 2014 байтов; добавьте 8 байтов, потому что существует две метки, настроенные под подчиненным интерфейсом.

При настройке **mtu 2026 байтов** под подчиненным интерфейсом он не применен, потому что это больше, чем MTU подчиненного интерфейса, наследованного от основного интерфейса

(2022). В результате можно настроить только MTU подчиненного интерфейса ниже, чем 2022 байта.

На основе этого MTU подчиненного интерфейса вычислите MTU информационного наполнения LDP MPLS, которое сообщено соседнему узлу, и удостоверьтесь, что это идентично тому, вычисленному удаленным PE L2VPN. Это - то, где команда **перезаписи** играет роль.

Для вычислений MTU информационного наполнения LDP MPLS возьмите MTU подчиненного интерфейса, тогда:

1. Вычтите 14 байтов для Заголовка ethernet.
2. Вычтите 4 байта для каждой метки, вытолканной в команде **перезаписи**, настроенной под подчиненным интерфейсом.
3. Добавьте 4 байта для каждой метки, выдвинутой в команде **перезаписи**, настроенной под подчиненным интерфейсом.

Это - тот же пример с конфигурацией QinQ на концерте 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)
```

Это - вычисление для MTU информационного наполнения LDP MPLS:

1. Значение MTU MTU подчиненного интерфейса: 2022 байта
2. Вычтите 14 байтов Заголовка ethernet: $2022 - 14 = 2008$ байтов
3. Вычтите 4 байта для каждой вытолканной метки в команде **перезаписи**: $2008 - 4 * 2 = 2000$

Гарантируйте, что удаленная сторона объявляет информационное наполнение LDP MPLS 2000 байтов. В противном случае отрегулируйте локальный или удаленный максимальный размер передаваемого блока данных прикрепляемого канала (AC), таким образом, они совпадают.

```
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
```

Определенные команды Encapsulation Точки потока Ethernet (EFP)

Эти инкапсуляции рассчитывают как совпадающий с нулевыми метками, таким образом, они не увеличивают MTU подчиненного интерфейса:

- **encapsulation untagged**
- **encapsulation default**

Эти модификаторы инкапсуляции не влияют на количество меток, требуемых для вычислений MTU подчиненного интерфейса:

- собственный компонент
- информационное-наполнение-ethertype
- точный
- потому что
- входной исходный Mac или входной целевой Mac

инкапсуляция [dot1q|dot1ad] помеченное приоритетом количество как соответствие с одиночной меткой.

'Любое' ключевое слово, используемое в качестве самого внутреннего соответствия метки, не увеличивает MTU подчиненного интерфейса.

- **encapsulation dot1q любой** не увеличивает MTU подчиненного интерфейса.
- **инкапсуляция dot1ad 10 dot1q любой** считается как одна метка; это увеличивает MTU подчиненного интерфейса на 4 байта.
- **инкапсуляция dot1ad любой dot1q 7** считается как две метки; это увеличивает MTU подчиненного интерфейса на 8 байтов.

Диапазоны VLAN-ID инкрементно увеличивают MTU подчиненного интерфейса:

- **encapsulation dot1q 10-100** считается как одна метка; это увеличивает MTU подчиненного интерфейса на 4 байта.

Исдержки MTU инкапсуляции EFP, который является дизъюнктивным соответствием, рассматриваются как MTU его самого высокого элемента.

- **encapsulation dot1q 10-100, без меток** считается как одна метка, потому что диапазон 10 - 100 является самым высоким элементом.

Non EVC (XR 12000 и CRS)

Маршрутизаторы как маршрутизатор Cisco серии XR 12000 и система маршрутизации Cisco операторского класса используют традиционную конфигурацию для VLAN, совпадающей на подчиненных интерфейсах. Эти характеристики применяются к интерфейсам L2VPN L2 на CRS и на XR 12000 маршрутизаторов, которые не придерживаются модели EVC:

- На non-EVC платформах автоматически разделены входящий dot1q или метки dot1ad, когда они получены на подчиненном интерфейсе транспорта L2.
- При вычислениях объема полезных данных для LDP MPLS для сигнализации, вычитите размер меток от MTU подчиненного интерфейса, как замечено в **команде show interface**.
- Это подобно случаю маршрутизовавшего подчиненного интерфейса.
- Подчиненный интерфейс наследовал свой MTU от основного интерфейса; добавьте 4 байта для каждой метки к MTU основного интерфейса для вычислений MTU подчиненного интерфейса. Например, если подчиненный интерфейс QinQ имеет 2

метки dot1q, подчиненный интерфейс, по умолчанию, имеет MTU, который на 8 байтов больше, чем MTU основного интерфейса.

- Можно также использовать команду **mtu** под подчиненным интерфейсом, но это используется только для сокращения MTU подчиненного интерфейса, который наследован от MTU основного интерфейса.

Вот несколько примеров, которые иллюстрируют эти характеристики.

Данный пример показывает, как настроен non-EVC подчиненный интерфейс:

```
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#
```

non-EVC платформы используют **dot1q vlan** или **dot1ad** команды **vlan** вместо инкапсуляции и переписывают команды платформ EVC (ASR9000).

Если вы не настраиваете MTU явно на основном или подчиненном интерфейсе, то 1500-байтовый пакет L3 может быть получен по умолчанию:

```
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#
```

MTU подчиненного интерфейса вычислен из MTU основного интерфейса (1514); добавьте 4 байта для каждой метки dot1q. Поскольку существует одна метка, настроенная на подчиненном интерфейсе с **dot1q vlan 201** команда, добавьте 4 байта к 1514 для MTU 1518 байтов.

Соответствующий MTU информационного наполнения в LDP MPLS составляет 1500 байтов, так как 14 байтов Заголовка ethernet не посчитаны, и одна метка dot1q вытолкана автоматически non-EVC платформой, когда это пробегается через 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
```

При увеличении MTU основного интерфейса к 2014 байтам MTU подчиненного интерфейса увеличен соответственно:

```
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
```

Так, для вычислений MTU LDP MPLS вычитите 14 байтов Заголовка ethernet и добавьте 4 байта для каждой метки, настроенной под подчиненным интерфейсом.

Автоматический MTU драйвера интерфейса Ethernet и конфигурация MRU

На Интерфейсах Ethernet интерфейсный драйвер настроен с MTU и MRU, который основывается на конфигурации максимального размера блока данных (MTU) интерфейса.

Настроенный MTU и MRU на Драйвере интерфейса Ethernet могут быть замечены с **<interface>** контроллера показа вся команда.

В версиях ранее, чем Выпуск 5.1.1 Cisco IOS XR, MTU и MRU на Драйвере интерфейса Ethernet были автоматически настроены на основе конфигурации MTU Cisco IOS XR на интерфейсе.

MTU/MRU, настроенный на Драйвере Ethernet, был просто на основе настроенного MTU + 12 байтов для добавления 2 Меток Ethernet и поля контрольной суммы. 12 байтов были добавлены к MTU/MRU Драйвера Ethernet, независимо от того, если были какие-либо теги VLAN, настроенные на подчиненных интерфейсах.

Пример со всеми версиями Cisco IOS XR ранее, чем Выпуск 5.1.1 Cisco IOS XR и MTU по умолчанию 1514 на интерфейсе ASR 9000 показывают здесь:

```
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>
```

В Выпуске 5.1.1 Cisco IOS XR и позже, MTU и MRU, который используется на Драйвере интерфейса Ethernet, изменился и теперь на основе количества тегов VLAN, которые настроены на любом из подчиненных интерфейсов.

Если никакие теги VLAN не настроены ни на каких подчиненных интерфейсах, MTU/MRU драйвера равняется настроенному MTU на интерфейсе + 4 байта CRC, например $1514 + 4 = 1518$ байтов.

Если одна VLAN настроена на каких-либо подчиненных интерфейсах, MTU/MRU драйвера равняется настроенному MTU + 8 байтов (1 метка + CRC), например $1514 + 8 = 1522$ байта.

Если два тега VLAN настроены на каких-либо подчиненных интерфейсах, MTU/MRU драйвера равняется настроенному MTU + 12 байтов (2 метки + CRC), например $1514 + 12 = 1526$ байтов

Если QinQ с **любым** ключевым словом настроен для второй-dot1q метки, MTU/MRU драйвера равняется настроенному MTU + 8 байтов (1 метка + CRC), например $1514 + 8 = 1522$ байта.

Эти примеры отображают поведение в Выпуске 5.1.1 Cisco IOS XR и позже 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>
```

В большинстве ситуаций это изменение поведения в Выпуске 5.1.1 и позже не должен требовать никаких изменений к конфигурации MTU на интерфейсе.

Это изменение поведения может вызвать проблемы в случае подчиненного интерфейса, настроенного с одиночным тегом VLAN , но получает пакеты с двумя тегами VLAN. В той ситуации полученные пакеты могут превысить MRU на Драйвере интерфейса Ethernet. Для устранения того условия максимальный размер блока данных (MTU) интерфейса может или быть увеличен на 4 байта или подчиненный интерфейс, настроенный с двумя тегами VLAN.

Автоматический MTU Драйвера интерфейса Ethernet и конфигурация MRU в поведении Выпуска 5.1.1 являются тем же для CRS и ASR 9000 маршрутизаторов. Но маршрутизатор CRS, который выполняет Выпуск 5.1.1, не включает 4-байтовый CRC в MTU и значение MRU, отображенное в **выходных данных show controller**. Поведение того, как об этом сообщают, не является тем же между CRS и 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)
```

Путем MTU и MRU отображены в выходных данных show controller на ASR 9000, изменится в будущем так, чтобы 4 байта CRC не были включены в отображенное значение MTU/MRU. Это будущее изменение может быть отслежено с идентификатором ошибки Cisco [CSCuo93379](https://www.cisco.com/cisco/web/errata/CSCuo93379).

Преобразуйте конфигурацию, когда вы обновите от выпуска ранее, чем выпуск 5.1.1 к выпуску 5.1.1 или позже

- MTU по умолчанию:

Если был основной интерфейс без какого-либо подчиненного интерфейса и без какой-либо команды `mtu` в выпуске ранее, чем Выпуск 5.1.1:

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

И этот интерфейсный dot1q транспортов или Кадры QinQ, тогда MTU должен быть вручную настроен к "mtu 1522" в Выпуске 5.1.1 и позже:

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

Эта конфигурация позволяет Кадрам QinQ транспортироваться как в более ранних релизах. Если только dot1q и не QinQ должен быть транспортирован, значение MTU могло быть настроено к 1518.

Если были подчиненные интерфейсы, настроенные для dot1q или QinQ, но с "любым" ключевым словом и никакими подчиненными интерфейсами QinQ с 2 явными метками был настроен в выпуске ранее, чем Выпуск 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
!
```

Эта конфигурация в Выпуске 5.1.1 и позже только позволит транспортным кадрам с одной меткой, таким образом, MTU должен будет также быть увеличен вручную на 4 байта, если должны быть транспортированы Кадры QinQ:

```
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
!
```

Если подчиненный интерфейс QinQ с 2 явными метками (которые не используют ключевое слово) настроен, нет никакой потребности модифицировать конфигурацию MTU, когда вы обновляете к Выпуску 5.1.1 и позже:

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

Если бы существует подчиненный интерфейс транспорта № L2, но только маршрутизируемые интерфейсы L3, ожидается, что конфигурация MTU совпала бы в обеих сторонах и не будет кадров, больше, чем MTU, который транспортируется. Когда вы обновляете к Выпуску 5.1.1 и позже, нет никакой потребности обновить конфигурацию MTU.

- MTU не по умолчанию в выпуске ранее, чем Выпуск 5.1.1:

Точно так же, когда MTU не по умолчанию был настроен в выпуске ранее, чем Выпуск 5.1.1 и никакой подчиненный интерфейс были настроены и dot1q, или Кадры QinQ должны быть транспортированы, тогда настроенная стоимость MTU должна быть увеличена на 8 байтов, когда вы обновляете к Выпуску 5.1.1 или позже.

Выпуск ранее, чем Выпуск 5.1.1:

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

Когда вы обновляете к Выпуску 5.1.1 и позже, MTU должен быть вручную увеличен на 8 байтов:

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

Если существует подчиненный интерфейс dot1q и никакой подчиненный интерфейс QinQ или подчиненный интерфейс QinQ с любым ключевым словом для метки второго dot1q, настроенная стоимость MTU должна также быть увеличена на 4 байта.

Выпуск ранее, чем Выпуск 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
!
```

Выпуск 5.1.1 и позже:

```
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
!
```

Если подчиненный интерфейс QinQ с 2 явными метками (которые не используют ключевое слово) настроен, нет никакой потребности модифицировать конфигурацию MTU, когда вы обновляете к Выпуску 5.1.1 и позже.

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

Если бы существует подчиненный интерфейс транспорта № L2, но только маршрутизируемые интерфейсы L3, ожидается, что конфигурация MTU совпала бы в обеих сторонах и не будет кадров, больше, чем MTU, который транспортируется. Когда вы обновляете к Выпуску 5.1.1 и позже, нет никакой потребности обновить конфигурацию MTU.