Cisco IOS XR和Cisco IOS路由器上的MTU行为

目录

简介

背景信息

- Cisco IOS和Cisco IOS XR软件的比较
- 路由的L3接口
  - 默认MTU
  - 非默认MTU
- 路由的L3子接口
- L2VPN L2接口
  - EVC(ASR9000)
  - 非EVC(XR 12000和CRS)

自动以太网接口驱动程序MTU和MRU配置

- 从版本5.1.1之前的版本升级到版本5.1.1或更高版本时，转换配置

简介

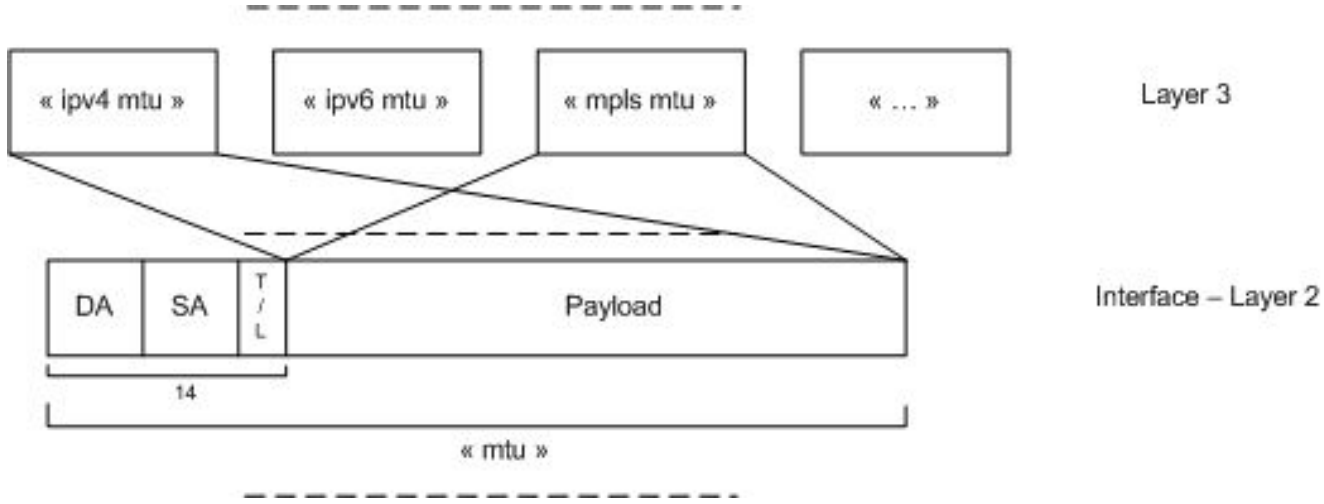
本文档介绍了Cisco IOS® XR路由器上的最大传输单元(MTU)行为，并将这些行为与Cisco IOS路由器进行比较。它还讨论了路由第3层(L3)接口和第2层VPN(L2VPN)第2层接口上同时使用以太网虚拟连接(EVC)和非EVC型号的MTU。本文档还介绍了在版本5.1.1及更高版本中如何自动配置以太网接口驱动程序MTU和最大接收单元(MRU)的重要更改。

背景信息

在计算机网络中，某层通信协议的MTU定义了允许该层通过一个接口传输的最大协议数据单元的大小（以字节为单位）。一个MTU参数与每个接口、层和协议关联。

Cisco IOS XR软件中的MTU特征如下：

- 第2层和第3层的MTU config和show命令包括其对应层的报头大小。例如，配置L2 MTU的mtu命令包括以太网接口的14个字节（不含dot1q），或点对点协议(PPP)或高级数据链路控制(HDLC)的4个字节。 ipv4 mtu命令包含20字节的IPv4报头。
- 较高层的MTU必须适合较低层的负载。例如，如果非dot1q以太网接口的接口MTU默认为1514字节，则更高层协议（如多协议标签交换[MPLS]）在该接口上的最大MTU可以为1500字节。这意味着您只能将1500字节的MPLS帧（包括标签）放入以太网帧中。如果想要在1500字节的IPv4数据包上允许两个MPLS标记，则不能在该接口上配置1508字节的MPLS MTU。为了在以太网接口上传输1508字节的MPLS帧，接口MTU必须增加到1522或更高的值，以确保第2层接口负载足够大以承载MPLS帧。



- 在经典的Cisco IOS软件（不是Cisco IOS XR软件）中，interface mtu命令配置L2负载大小，但不包括L2报头。这与Cisco IOS XR软件不同，Cisco IOS XR软件在interface mtu命令中同时包含L2和L3开销。L3 MTU命令(对于ipv4 mtu命令)可配置包含L3报头的协议的最大数据包大小。这类似于Cisco IOS XR软件。
- Cisco IOS XR软件中的默认接口MTU必须允许传输1500字节的L3数据包。因此，主以太网接口的默认MTU为1514字节，串行接口的默认MTU为1504字节。

本文档的其余部分说明MTU特征，比较Cisco IOS和Cisco IOS XR软件行为，并提供以下类型接口的示例：

- 路由的L3接口
- 路由的L3子接口
- L2VPN L2接口

Cisco IOS和Cisco IOS XR软件的比较

本部分将Cisco IOS和Cisco IOS XR软件行为与MTU特征进行比较。

在Cisco IOS软件中，mtu命令和相应的show命令不包含L2报头。使用mtu命令将L2负载配置为第3层数据包（包括L3报头）的最大大小。

这与Cisco IOS XR软件不同，其中mtu命令包括L2报头（以太网为14字节，PPP/HDLC为4字节）。

如果Cisco IOS路由器配置了mtu x并连接到Cisco IOS XR路由器，则Cisco IOS XR路由器上的相应接口应配置有mtu x+14（用于以太网接口）或mtu x+4（用于串行接口）。

Cisco IOS和Cisco IOS XR软件对ipv4 mtu、ipv6 mtu和mpls mtu命令具有相同的含义；它们必须配置相同的值。

因此，这是以太网接口上Cisco IOS软件中的配置：

```
mtu 9012
ipv4 mtu 9000
```

```
ipv6 mtu 9000
```

Cisco IOS XR软件邻居上的相应配置为：

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

路由的L3接口

连接到第2层网络的所有设备上的MTU值必须相同。否则，可能会报告以下症状：

- 中间系统到中间系统(IS-IS)邻接关系不会出现。默认情况下，IS-IS使用hello-padding;因此，hello可能被视为巨型帧，当一台路由器的MTU值低于其他路由器的MTU值时，可能会将其丢弃。
- 开放最短路径优先(OSPF)邻接关系会停滞在Exstart或Exchange状态，因为大型数据库描述符(DBD)数据包可能会被定义为巨型数据包，可能会被丢弃。当路由器收到具有较低MTU值的数据包时，数据库不会同步。
- 数据流量的特征是巨型数据流，当数据流量在传输设备上接收时，其MTU值低于传输设备上的MTU值，数据流量将被丢弃。
- 大数据包被丢弃时，吞吐量很低。对于路径MTU发现，TCP会话可以在丢弃大数据包时恢复，但这会影响吞吐量。

默认MTU

本节分析未配置mtu命令时路由接口的默认MTU:

<#root>

```
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_GETHERNET
  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#
```

在本示例中，默认L2接口MTU为1514字节，包括14字节的以太网报头。14个字节由目的MAC地址的6个字节、源MAC地址的6个字节以及类型或长度的2个字节计算。这不包括前导码、帧定界符、4字节的帧校验序列(FCS)和帧间间隙。对于PPP或HDLC帧，L2报头包含4个字节；因此默认接口MTU为1504字节。

L3子协议从父MTU的有效负载继承其MTU。当您从1514字节的L2 MTU中减去L2报头的14字节时，您的L2负载为1500字节。这成为L3协议的MTU。IPv4、IPv6、MPLS和无连接网络服务(CLNS)继承此1500字节的MTU。因此，默认情况下，Cisco IOS XR以太网接口可以传输1500字节的第3层数据包，该数据包与Cisco IOS以太网接口上的default相同。

非默认MTU

本部分显示如何配置mpls mtu 1508，以便在数据包顶部发送1500字节的IPv4数据包，其中每个IPv4数据包具有两个4字节的MPLS标记：

```
<#root>
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_GETHERNET
  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#
```

虽然mpls mtu 1508命令已提交，但并未应用，因为MPLS在show命令中仍有1500字节的MTU。这是因为L3子协议的MTU不能大于其父L2接口的负载。

要在1500字节的IP数据包上允许两个标签，您必须：

- 配置一个1522字节的第2层接口MTU，以便所有子协议（包括MPLS）继承1508字节的MTU(1522 - 14 = 1508)。
- 将L3协议的MTU减少到1500字节，以便仅允许MPLS超过1500字节。

<#root>

```
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_GETHERNET
  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#
```

通过此配置，您可以发送1500字节的IPv4和IPv6数据包以及1508字节的MPLS数据包（顶部有两个标记的1500字节数据包）。

路由的L3子接口

这些特征适用于路由的L3子接口。

路由的子接口MTU继承其父主接口的MTU;为子接口上配置的每个VLAN标记添加4个字节。因此，dot1q子接口有4个字节，IEEE 802.1Q隧道(QinQ)子接口有8个字节。

因此，相同大小的L3数据包可以在主接口和子接口上转发。

可以在子接口下配置mtu命令，但仅当该命令低于或等于从主接口继承的MTU时才应用。

以下示例中主接口的MTU为2000字节：

<#root>

```
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#
```

在show命令中，子接口的MTU为2004;在主接口的MTU中添加4个字节，因为子接口下配置了一个dot1q标记。

但是，IPv4和IPv6数据包的MTU仍然与主接口(1986)的MTU相同。 这是因为L3协议的MTU现在计算为：2004 - 14 - 4 = 1986。

可以在子接口下配置mtu命令，但配置的MTU仅在小于或等于从主接口继承的MTU（比主接口的MTU大4字节）时才应用。

当子接口的MTU大于继承的MTU时，不会应用它，如下所示：

<#root>

```
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#
```

因此，您只能使用mtu命令来降低从主接口继承的MTU值。

同样，您还可以使用L3协议(IPv4、IPv6、MPLS)的MTU命令来降低从子接口L2负载继承的L3 MTU值。当将L3协议MTU配置为不适合L2 MTU负载的值时，MTU不起作用。

L2VPN L2接口

L2VPN的MTU非常重要，因为当PW两端的连接电路上的MTU不同时，标签分配协议(LDP)不会启动伪线(PW)。

以下是show命令，该命令说明L2VPN PW在MTU不匹配时停机：

<#root>

```
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 0xfffe0001
    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#
```

在本示例中，请注意，每一端的MPLS L2VPN提供商边缘(PE)必须发出相同的MTU值才能启用PW。

由MPLS LDP发出信号的MTU不包括L2开销。这不同于包含L2开销的XR interface config和show命令。子接口上的MTU为2018字节（继承自2014字节的主接口），但LDP发出的MTU为2000字节。因此，它从L2报头中减去18字节（14字节的以太网报头+ 4字节的1 dot1q标记）。

了解每台设备如何计算连接电路的MTU值以修复MTU不匹配非常重要。这取决于供应商、平台、软件版本和配置等参数。

EVC(ASR9000)

Cisco ASR 9000系列聚合服务路由器使用EVC基础设施模型，允许在L2VPN L2接口和子接口上灵活地匹配VLAN。

EVC L2VPN L2接口具有以下特征：

- 它们允许使用encapsulation命令配置一个或多个标记。
- 默认情况下，仅使用encapsulation命令即可通过PW保留和传输标记。因此，在默认情况下，您不需要像在非EVC平台上那样移除标记。
- 当您决定弹出传入的标记或将一些附加标记推送到传入帧顶部时，请使用rewrite命令。

要计算子接口MTU，请采用主接口MTU（默认值或手动在主接口下配置的），并为使用encapsulation命令配置的每个VLAN标记添加4个字节。

当子接口下有mtu命令时，仅当该命令低于计算MTU时才生效。rewrite命令不会影响子接口的MTU。

例如：

<#root>

```
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)
```

在本示例中，主接口中的MTU为2014字节；添加8个字节，因为子接口下配置了两个标记。

如果在子接口下配置mtu 2026字节，则不会应用它，因为它大于从主接口(2022)继承的子接口的MTU。因此，只能配置小于2022字节的子接口MTU。

根据此子接口MTU，计算向邻居发出信号的MPLS LDP负载的MTU，并确保它与远程L2VPN PE计算的负载相同。此时将执行rewrite命令。

为了计算MPLS LDP负载的MTU，请采用子接口的MTU，然后：

1. 为以太网报头减去14个字节。
2. 为子接口下配置的rewrite命令中弹出的每个标记减去4个字节。
3. 为子接口下配置的rewrite命令中推送的每个标记添加4个字节。

以下是gig 0/1/0/3.201上QinQ配置的相同示例：

<#root>

```
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)
```

以下是MPLS LDP负载MTU的计算：

1. 子接口的MTU值：2022 bytes
2. 减去14字节的以太网报头：2022 - 14 = 2008字节
3. 在rewrite命令中为每个弹出标记减去4个字节：2008 - 4 * 2 = 2000

确保远程端通告2000字节的MPLS LDP负载。否则，请调整本地或远程连接电路(AC)的MTU大小，使其匹配。

```
<#root>

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
```

特定以太网流点(EFP)封装命令

这些封装会计为匹配的零标记，因此它们不会增加子接口MTU:

- encapsulation untag
- 默认封装

这些封装修饰符不影响计算子接口MTU所需的标记数量：

- 本地
- payload-ethertype
- 精确

- cos
- 入口源mac或入口目标mac

encapsulation [dot1q|dot1ad] priority-tagged计数与单个标记匹配。

用作最内部标记匹配的“any”关键字不会增加子接口MTU。

- encapsulation dot1q any不会增加子接口MTU。
- encapsulation dot1ad 10 dot1q any作为单个标记；它将子接口MTU增加4个字节。
- encapsulation dot1ad any dot1q 7作为两个标签；它将子接口MTU增加8个字节。

VLAN-ID的范围会增加子接口MTU:

- encapsulation dot1q 10-100记为一个标记；它将子接口MTU增加4个字节。

作为分离匹配的EFP的封装MTU开销被视为其最高元素的MTU。

- encapsulation dot1q 10-100, untagged作为一个标记，因为10-100范围是最高元素。

非EVC(XR 12000和CRS)

Cisco XR 12000系列路由器和运营商路由系统(CRS)等路由器使用子接口上VLAN匹配的传统配置。以下特征适用于不遵循EVC模型的CRS和XR 12000路由器上的L2VPN L2接口：

- 在非EVC平台上，在L2传输子接口上收到传入的dot1q或dot1ad标记时，这些标记将被自动剥离。
- 当您计算要发出信号的MPLS LDP的负载大小时，从子接口的MTU中减去标记的大小，如show interface命令所示。
- 这与路由子接口的情况类似。
- 子接口从主接口继承其MTU;将每个标记的4个字节添加到主接口的MTU以计算子接口的MTU。例如，如果QinQ子接口有2个dot1q标记，则默认情况下，该子接口的MTU比主接口的MTU大8个字节。
- 您也可以在子接口下使用mtu命令，但它仅用于减少子接口的MTU，从主接口的MTU继承。

这里有几个例子说明了这些特征。

此示例说明如何配置非EVC子接口：

<#root>

```
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#
```

非EVC平台使用dot1q vlan或dot1ad vlan命令，而不是EVC平台(ASR9000)的encapsulation和rewrite命令。

如果未在主接口或子接口上显式配置MTU，则默认情况下可以接收1500字节的第3层数据包：

```
<#root>

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#
```

子接口MTU从主接口MTU(1514)计算；为每个dot1q标记添加4个字节。由于子接口上使用dot1q vlan 201命令配置了一个标记，因此对于1518字节的MTU，请将4个字节添加到1514。

MPLS LDP中相应的负载MTU为1500字节，因为以太网报头的14字节不计数，并且，当非EVC平台通过PW时，会自动弹出1 dot1q标记：

```
<#root>

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
```

如果将主接口的MTU增加到2014字节，子接口的MTU将相应地增加：

```
<#root>

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
```

因此，为了计算MPLS LDP MTU，请减去14字节的以太网报头，并为子接口下配置的每个标记添加4个字节。

自动以太网接口驱动程序MTU和MRU配置

在以太网接口上，接口驱动程序配置有基于接口MTU配置的MTU和MRU。

使用show controller <interface> all命令可以查看以太网接口驱动程序上配置的MTU和MRU。

在早于Cisco IOS XR版本5.1.1的版本中，以太网接口驱动程序上的MTU和MRU是根据接口上的Cisco IOS XR MTU配置自动配置的。

在以太网驱动程序上配置的MTU/MRU仅基于配置的MTU + 12字节，用于添加2个以太网标记和CRC字段。无论子接口上是否配置了VLAN标记，12个字节都已添加到以太网驱动程序MTU/MRU中。

以下示例显示了所有Cisco IOS XR版本（低于Cisco IOS XR版本5.1.1）和ASR 9000接口上的默认MTU 1514:

<#root>

```
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>
```

在Cisco IOS XR版本5.1.1及更高版本中，以太网接口驱动程序中使用的MTU和MRU已更改，现在基于任何子接口上配置的VLAN标记数量。

如果在任何子接口上未配置VLAN标记，则驱动程序MTU/MRU等于接口上配置的MTU + 4 CRC字节，例如1514 + 4 = 1518字节。

如果在任何子接口上配置了一个VLAN，则驱动程序MTU/MRU等于配置的MTU + 8字节（1标记+ CRC），例如1514 + 8 = 1522字节。

如果在任何子接口上配置了两个VLAN标记，则驱动程序MTU/MRU等于配置的MTU + 12字节（2个标记+ CRC），例如1514 + 12 = 1526字节

如果为second-do1q标记配置带有any关键字的QinQ，则驱动程序MTU/MRU等于配置的MTU + 8字节（1标记+ CRC），例如1514 + 8 = 1522字节。

以下示例显示了ASR 9000上Cisco IOS XR版本5.1.1及更高版本中的行为：

<#root>

```
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>
```

在大多数情况下，版本5.1.1及更高版本中的此行为更改不应要求对接口上的MTU配置进行任何更改。

如果子接口配置了单个VLAN标记，则此行为更改会导致问题，但会收到具有两个VLAN标记的数据包。在这种情况下，收到的数据包可能会超过以太网接口驱动程序上的MRU。为了消除这种情况，接口MTU可以增加4个字节，也可以使用两个VLAN标记配置子接口。

5.1.1版本中的自动以太网接口驱动程序MTU和MRU配置对于CRS和ASR 9000路由器而言是相同的。但是，运行版本5.1.1的CRS路由器在show controller输出中显示的MTU和MRU值中不包含4字节的CRC。 CRS和ASR9000之间的报告行为不同。

<#root>

```
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)
```

ASR 9000上show controller输出中MTU和MRU的显示方式将来会更改，因此所显示的MTU/MRU值中不会包含CRC的4个字节。可使用Cisco Bug ID CSCuo93379跟踪此将来的更改。

从版本5.1.1之前的版本升级到版本5.1.1或更高版本时，转换配置

- 默认MTU:

如果5.1.1之前的版本中存在不带任何子接口且没有mtu命令的主接口：

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

并且此接口传输dot1q或QinQ帧，然后，在版本5.1.1及更高版本中，应将MTU手动配置为“mtu

1522”：

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

此配置允许QinQ帧像较早版本一样进行传输。如果只传输dot1q而不传输QinQ，则MTU值可配置为1518。

如果存在为dot1q或QinQ配置的子接口，但带有“any”关键字，且版本低于5.1.1的版本中未配置带有2个显式标记的QinQ子接口：

```
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
!
```

版本5.1.1及更高版本中的此配置只允许传输带一个标记的帧，因此，如果要传输QinQ帧，MTU也应手动增加4字节：

```
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
!
```

如果配置了带有2个显式标记（不使用“any”关键字）的QinQ子接口，则在升级到版本5.1.1及更高版本时，无需修改MTU配置：

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

如果没有L2传输子接口，而只有L3路由接口，则预计两端的MTU配置将会匹配，并且不会出现大于传输的MTU的帧。升级到版本5.1.1及更高版本时，无需更新MTU配置。

- 早于5.1.1版本的非默认MTU:

同样，在早于5.1.1版本的版本中配置了非默认MTU且未配置任何子接口且必须传输dot1q或QinQ帧时，当您升级到版本5.1.1或更高版本时，配置的MTU值应增加8字节。

低于5.1.1的版本：

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

升级到版本5.1.1及更高版本时，应手动增加8个字节：

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

如果存在dot1q子接口且没有QinQ子接口或带有any关键字的second-dot1q子接口，则配置的MTU值也应增加4字节。

低于5.1.1的版本：

```
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
!
```

版本5.1.1及更高版本：

```
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
!
```

如果配置了2个显式标记（不使用“any”关键字）的QinQ子接口，则升级到版本5.1.1及更高版本时，无需修改MTU配置。

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

如果没有L2传输子接口，但只有L3路由接口，则预计两端的MTU配置将匹配，并且不会出现大于传输的MTU的帧。升级到版本5.1.1及更高版本时，无需更新MTU配置。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。