

MPLS MTU

Maximum transmission unit (MTU) is the maximum size, in bytes, of a packet or frame transmitted over a network. This chapter talks about MPLS MTU configuration for MPLS enabled interfaces.

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
MPLS MTU	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.
MPLS MTU	Release 7.10.1	You can now enforce the size of MPLS frames in your MPLS network by configuring the MPLS MTU of the MPLS interface. This enables interworking with routers from other vendors that have a strict MPLS MTU enforcement. When a transit packet exceeds the MPLS MTU, the router either fragments the packet or drops it. Cisco 8000 Series Routers and Line Cards with Cisco Silicon One Q200, Q201, or Q202 ASIC support this feature. In earlier releases, Cisco 8000 Series Routers didn't support MPLS MTU feature. While interworking with other routers that had MPLS MTU configured, the Cisco 8000 Series Router sent packets exceeding the MPLS MTU of those router interfaces. This resulted in packet drops at the peer routers. You can configure MPLS MTU with either CLI or YANG data model: **CLI:** • **mpls mtu** **YANG Data Model:** • `Cisco-IOS-XR-um-if-mpls-cfg.yang` (see GitHub, YANG Data Models Navigator)

• Overview of MPLS MTU, on page 2
• Configuration Example for MPLS MTU, on page 4

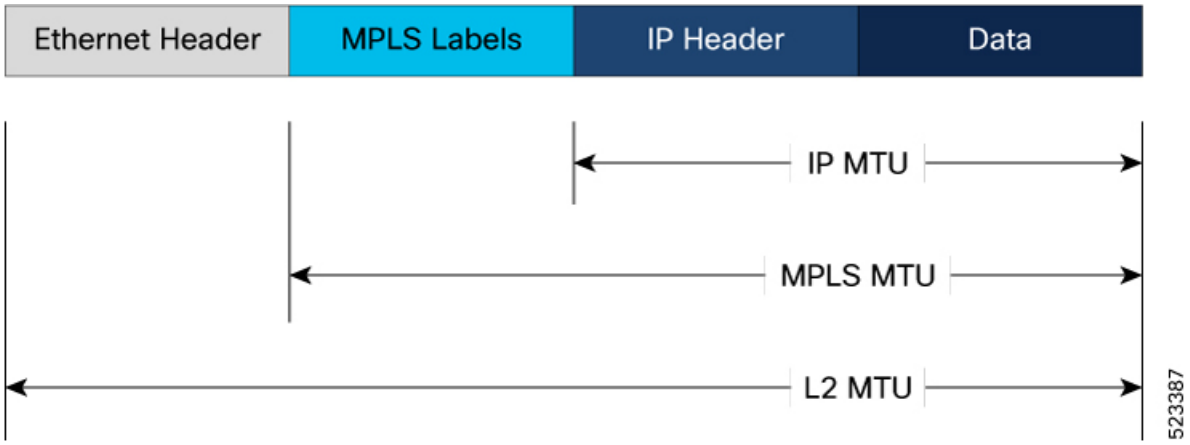
Overview of MPLS MTU

The MTU of an interface determines whether a packet is too large to be transmitted across it. You can configure three types of MTU values on a router interface:

- Layer 2 (L2) or Ethernet MTU which specifies the maximum transmittable size of the packet, measured from the ethernet header until the end of the packet. The default value of L2 MTU for a main interface is 1514 bytes. This value is configurable with the **mtu** command.
- IP MTU which specifies the maximum transmittable size of the packet, measured from the IP header until the end of the packet. This value is applicable only for unlabeled IPv4 or IPv6 packets. The default value of IP MTU is L2 MTU subtracted by 14 bytes which is the size of the ethernet header of the main interface. You can configure the IP MTU with the **ip mtu** command.
- MPLS MTU which specifies the maximum transmittable size of the packet measured from the MPLS labels until the end of the packet. This value is applicable only for labeled packets. The default value of MPLS MTU is L2 MTU subtracted by 14 bytes which is the size of the ethernet header of the main interface. You can configure the MPLS MTU with the **mpls mtu** command.

The following figure depicts the different portions of the egress packet that the router considers for MTU checks.

Figure 1: Different portions of the egress packet that the router considers for MTU checks



In an IPv4 network with MPLS enabled, the Label Switching Routers add one or more MPLS labels to all IPv4 packets. Each label is 4 bytes. So, a packet with *n* number of labels, would have *n* x 4 bytes added to it. Consider that the maximum size of an IPv4 packet on this network is 1592 bytes. If the router imposes a maximum of two labels on all IPv4 packets transmitted, then you can set the following values for L2 MTU and MPLS MTU:

- L2 MTU = 1592 + size of two MPLS labels + size of the ethernet header = 1592 + (2 x 4) + 14 = 1614 bytes
- MPLS MTU = 1592 + size of two MPLS labels = 1592 + (2 x 4) = 1600 bytes

Configuring these values ensures that the router can transmit all IPv4 packets without fragmentation or packet drops.

In cases where the MPLS frame size exceeds the MPLS MTU of the interface, the router sends the packet to the LC CPU for exception processing. Exception processing depends on whether the underlying packet is IPv4 or IPv6.

If an MPLS frame exceeds MPLS MTU and has an underlying IPv4 packet, then the LC CPU fragments or drops the packet based on the Don't Fragment (DF) bit. DF bit is a bit within the IP header that determines whether a router can fragment a packet. The packet originator usually sets the DF bit. But it's possible for any device in the path to set it too.

- If the DF bit isn't set, the LC CPU of the router fragments the packet. After fragmentation, the LC CPU attaches the MPLS header on each of the fragments, and then forwards each fragment to the destination. Since all fragments require their own header, it adds bandwidth overheads and is not desirable.
- If the DF bit is set, the LC CPU of the router drops the packet and generates ICMP unreachable messages which the router sends to the device where the packet originated.

If the MPLS frame that exceeded MPLS MTU has an underlying IPv6 packet, the LC CPU drops the packet and generates ICMPv6 packet-too-big messages which the router sends to the device where the packet originated.



Note There are two scenarios where the router applies a default value of MPLS MTU, that equals the L2 MTU subtracted by the ethernet header size of the main interface, which is 14 bytes:

- When you don’t configure the MPLS MTU for an MPLS enabled main interface or sub-interface.
- When you configure MPLS MTU to a value greater than the L2 MTU.

The **MTU** field in the output of the command **show interface** *interface-name* displays the L2 MTU of the interface.

```
Router#show interfaces fourHundredGigE 0/0/0/0
Thu Jul 27 05:04:54.955 UTC
FourHundredGigE0/0/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is FourHundredGigE, address is 78dd.c079.8400 (bia 78dd.c079.8400)
  Internet address is Unknown
  MTU 1514 bytes, BW 400000000 Kbit (Max: 400000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 400000Mb/s, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  Last link flapped 00:00:20
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
```

For more information on IP MTU and L2 MTU, refer the topics *IP MTU* and *Ethernet MTU* in the chapter *Configuring Ethernet Interfaces* in the *Interface and Hardware Component Configuration Guide for Cisco 8000 Series Routers*.

Restrictions for MPLS MTU

The following restrictions apply for MPLS MTU:

- Routers and line cards with the Cisco Silicon One Q200, Q201, or Q202 ASIC support the MPLS MTU feature. For more information on the routers and line cards with the Q200, Q201, or Q202 ASIC, refer *Cisco 8000 Series Routers Data Sheet*.
- If you configure MPLS MTU to a value greater than the L2 MTU, the router restricts the MPLS MTU to the value that equals the L2 MTU subtracted by the L2 header size of the main interface, which is 14 bytes.
- You can't configure MPLS MTU on Bridged Virtual Interfaces (BVI). Instead, the router applies a default MPLS MTU that equals the L2 MTU subtracted by the L2 header size of the BVI, which is 14 bytes.

Configuration Example for MPLS MTU

Configure **mpls mtu** command under **interface** configuration submode. The following configuration snippet sets the **mpls mtu** for the interface **FourHundredGigE 0/0/0/0** to 1400 bytes:

```
Router#config terminal
Router(config)#interface FourHundredGigE0/0/0/0
Router(config-if)#mpls mtu 1400
Router(config-if)#commit
```

Running Configuration

```
Router#show running-config interface FourHundredGigE0/0/0/0
interface FourHundredGigE0/0/0/0
 mpls
  mtu 1400
 !
!
```

Verification of MPLS MTU

The field starting with **mpls** or **Imposed MTU** in the output of the command **show im database interface** *interface-name* **detail | begin mpls** displays the configured value of MPLS MTU:

```
Router#show im database interface FourHundredGigE0/0/0/0 detail | begin mpls
  mpls             mpls (up, 1400)
    Views:                  LDP|G3P|L3P|OWN
    Owner Private:          144 bytes
    Instance ID:            -
    MPLS Label Stack Depth:
      Primary:              3
      Backup:               8
      SRTE:                 8
    Resource in NetIO:      TRUE
    NPU Mask (0/RP0/CPU0):  0x00000001 (1)
      ONLY IN VIEWS:          LDP
    Control flags:          0x00000030 (RDY|DNLD)
```

```
    Parent MTU:             1500
    Callback handle:        0x6001ff00
    Imposed MTU:            1400
    State constraint:       up
    Child MTU:              1400
  ether_sock      ether_sock (up, 1500)
    Views:                  LDP|G3P|L3P|OWN
    Owner Private:          144 bytes
    Instance ID:            -
    Resource in NetIO:      TRUE
    NPU Mask (0/RP0/CPU0):  0x00000001 (1)
      ONLY IN VIEWS:          LDP
    Control flags:          0x00000030 (RDY|DNLD)
    Parent MTU:             1500
    Callback handle:        0x6000bf00
    State constraint:       up
    Child MTU:              1500
```

You can also verify the MPLS MTU configuration of an MPLS-enabled interface with the command **show mpls interfaces** *interface-name* **private location** *location*.

, as shown:

```
Router#show mpls interfaces FourHundredGigE 0/0/0/0 private location 0/0/CPU0
Fri Nov  4 07:37:53.560 UTC
Interface      IFH         MTU   Label Stack Depth
                                  Prim Bkup Srte
-------------- ---------- ----- --------------------

FH0/0/0/0      0x01800220  1400     3    3    8
```