

Multilink Interfaces

Starting with Cisco IOS XE Amsterdam 17.1.x, the router supports configuring the multilink interfaces.

- MLPPP, on page 1
- MLPPP Interworking, on page 4
- MLPPP Layer 3 Termination, on page 8
- MLPPP ACR Support for iMSG, on page 9
- Multilink PPP Minimum Links Mandatory, on page 10
- Multilink Frame Relay, on page 11
- Multilink Frame Relay Layer 3 Termination, on page 16
- Multilink Frame Relay End-to-End Mode, on page 20

MLPPP

The Multilink Point-to-Point (MLPPP) feature enables you to create multilinks using an LCP option negotiation between two end points. The router A indicates to its peer router B that it is capable of combining multiple physical links into a logical bundle. Once the LCP negotiation is complete, the router A sends the multilink option to the router B and creates multilinks.

Starting with Cisco IOS XE 17.1.x, Multiprotocol Label Switching (MPLS) is supported over Multilink PPP (MLP) links in an access interface.

Restrictions for MLPPP

- MLPPP bundle is supported only on the RSP3 3G module.
- The MLPPP bundle should have member links only from the same interface module, if the member link is from different interface modules such as T1, E1, or OCx, then the MLPPP isn't supported.
- Each router supports up to a maximum of 1020 channels, hence a maximum number of bundles that can be configured is up to 510 until Cisco IOS XE Bengaluru 17.4.1.
- The change of encapsulation isn't supported if the member link is part of the multilink bundle.
- The maximum number of links in the bundle is 16.
- L3 termination isn't supported.

- The PPP options such as endpoint and member links are supported, and the options such as fragmentation, fragment-delay, inter-leave, idle link, multiclass, and change in MRRU value aren't supported.
- The MLPPP bundle can have either T1 or E1 links. It can't have a combination of T1 type and E1 type links.
- One MLPPP bundle can contain links belonging to a single interface module.
- The MLPPP bundle ID number that is supported is from 1 through 511 only.
- Fragmentation isn't supported.
- NCP, queue, and slippage aren't supported.
- Change in MRRU and MTU values on multilink interface isn't supported.
- The maximum number of multilink (max-link) configuration is supported in the MLPPP bundle. If the max-link value is configured as five, then the five links are active in the bundle. If one of five active links goes down, then there's no back-up link to become active.
- The member links of the MLPPP bundle can't be combined from different port types such as T1, T3, and SONET.
- Multiport isn't supported for MLPPP.

How to Configure MLPPP

Configuring MLPPP

To create the MLP interface, enter the following commands:

```
router(config)#interface multilink <group_number>
```



Note From release Cisco IOS XE Cupertino 17.9.1, the multilink group number range is restricted to only to 1–1024 for the following interface modules:

- NCS 4200 1-Port OC-192 or 8-Port Low Rate CEM 20G Bandwidth (NCS4200-1T8S-20CS)
- 1-port OC-48/STM-16 or 4-port OC-12/OC-3 / STM-1/STM-4 + 12-port T1/E1 + 4-port T3/E3 CEM (NCS4200-3GMS)

No changes in the multilink group number range for other interface modules, as it remains the same with 1–65535.

To add a PPP member to the channel, use the **ppp multilink group <group_number>** command. This command binds the serial interface with the corresponding multilink number (it typically adds the interface to the multilink). The PPP timeout retry should be added under the link if it's a part of the MLPPP ACR bundle.

The same configuration should be performed on the CE bundle links serial interface.

```
interface serial x/y/z.<channel_id>
    encapsulation ppp
```

```
ppp multilink group  <group_number>
ppp timeout retry 1
```

Two keys define the identity of a remote system: the PPP username and Multilink PPP endpoint discriminator. PPP authentication mechanisms (for example, password authentication protocol [PAP] or Challenge-Handshake Authentication Protocol [CHAP]) learn the PPP username. The endpoint discriminator is an option negotiated by the Link Control Protocol (LCP). Therefore, a bundle consists of links that have the same PPP usernames and endpoint discriminators.

The multilink group number is local to the system. However, for links to be a part of the bundle, the end-point discriminator helps to correctly identify which bundle the link needs to be part of. This end-point discriminator needs to be the same on both ends of the multilink.

```
interface serial<interface-name>
ppp multilink group <>
ppp multilink endpoint string <unique string>
```

The following example shows how to configure multilink interface:

```
interface multilink 30
ppp multilink endpoint string T1_multilink
interface serial0/3/4.1
encapsulation ppp
ppp multilink group 30
ppp timeout retry 1
```

Configuring MLPPP ACR

To create the MLP interface, enter the following commands:

```
router(config)#interface multilink <group_number>
```

To configure MLPPP ACR, enter the following commands:

The PPP timeout retry should be added under the link if it is a part of the MLPPP ACR bundle.

The same configuration should be performed on the CE bundle links serial interface.

```
serial-acr<group number.channel id>
encapsulation ppp
ppp multilink group  <group_number>
ppp timeout retry 1
```

The following example shows how to configure MLPPP ACR:

```
inter SERIAL-ACR3.1
encapsulation ppp
ppp multilink group 1
ppp timeout retry 1
```

For more information to create serial ACR interface, see iMSG ACR.

Deleting Serial Interface from an Active MLPPP Bundle

To remove a serial interface from the member link bundle, perform the following steps:

1. Shut the member link and ensure that the serial interface moves to down state on the other side of the bundle.

2. Remove the serial interface link from the MLPPP bundle using the **no ppp multilink group <group-id>** command.

Removing MLPPP Configuration

To delete the MLPPP bundle, you need to remove the serial interfaces linked to the bundle, and then remove link from the the serial interfaces.

To delete the MLPPP bundle, enter the following commands:

```
Router#configure terminal
router(config)#interface multilink <group_number>
router(config)#interface serial x/y/z.<channel_id>
router(config-if)#no ppp multilink group <multilink-id>
router(config-if)#no ppp multilink
```

To delete the multilink interface, you need to shut the multilink interface, and then delete the interface.

To delete the multilink interface, enter the following commands:

```
Router#config terminal
Router(config)#interface multilink <multilink-id>
Router(config-if)#shutdown
Router(config-if)#exit
Router(config)#no interface multilink <multilink-id>
```

Verifying MLPPP Configuration

To view the multilink bundle status and describes all the bundle parameter and details, use the **Show ppp multilink active** command:

```
router#Show ppp multilink active
Bundle name: rsp3-vz-pe1
  Remote Endpoint Discriminator: [1] rsp3-vz-pe1
  Local Endpoint Discriminator: [1] ASR_1000_CE_1
  Bundle up for 00:07:00, total bandwidth 1536, load 1/255
  Receive buffer limit 12000 bytes, frag timeout 1000 ms
  Bundle is Distributed
    0/0 fragments/bytes in reassembly list
    0 lost fragments, 0 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x53 received sequence, 0x0 sent sequence
  Platform Specific Multilink PPP info
    NOTE: internal keyword not applicable on this platform
    Interleaving: Disabled, Fragmentation: Disabled
  Member links: 1 active, 0 inactive (max 16, min not set)
    Se0/1/0.1/1/1:1, since 00:07:00
```

MLPPP Interworking

The MLPPP interworking supports IPv4 or IPv6 Layer 2 VPN Interworking with T1 or E1 bundles on the Cisco RSP3 module.

The MLPPP interworking enables service providers (offering relatively low-speed links) to use MLP and spread traffic across them in their MPLS networks. The MPLS Multilink PPP feature reduces the number of Interior Gateway Protocol (IGP) adjacencies and facilitates load sharing of traffic.

The MLPPP creates bundle in the following manner:

- T1 and E1 members should be from the same interface module.
- T1 and E1 members should be of the same port type. For example, if MLPPP link is from the T1, E1, or OCx ports, then only the links from the respective T1, E1, or OCx ports can be bundled together.
- Supports a maximum of 510 MLPPP pseudowires per interface module and 510 MLPPP pseudowires per router.

The following table details the number of bundles created in a module.

Table 1: MLPPP Bundle Created for Cisco RSP3 Module:

Number of Links in a Bundle	Number of VLANs Used	Maximum Number of Bundles in Cisco RSP3 Module
1	2	1020/2 = 510
2	3	1020/3 = 340
16	17	1020/17 = 60

Restrictions for MLPPP Interworking

- The router supports links equal to T1 or E1 for MLPPP bundling. You cannot bundle high-speed links (for example, E3) as the interface module can store only 50 ms of data based on the E1 speed.
- Each MLPPP bundle supports up to a maximum of 16 member links.
- Each MLP Bundle can have serial links from the same interface module or port type only.
- The MTU value supported is 1500 and MRRU should be more than the value of MTU, which is 1504.
- The jumbo frames are not supported.
- The keepalive is based on the first serial link attached to the MLP bundle.

How to Configure MLPPP Interworking

Configuring L2VPN Interworking and Cross Connection under the Attachment Circuit for Multilink Interfaces

Configuring L2VPN Interworking

To configure L2VPN interworking, create a pseudowire class with the tunneling encapsulation as MPLS. The **interworking** command specifies the type of payload traffic that flows across the pseudowire tunnel. Configure pseudowire class only once on a device.

You can also configure **control-word** as an optional command.

To configure L2VPN interworking, use the following commands:

```
router>enable
router#configure terminal
router(config)#interface pseudowire pw-number
router(config-if)# encapsulation mpls
router(config-if)# neighbor <peer-address><vcid-value>
router(config-xconnect)# control-word include
```

The following example shows how to configure L2VPN interworking:

```
interface pseudowire30
 encapsulation mpls
 neighbor 10.2.2.2 30
 control-word include
```

Configuring Cross-Connect Under Attachment Circuit

The **xconnect** command binds the attachment circuit to an L2VPN pseudowire for cross connect service.

The virtual circuit identifier creates the binding between a pseudowire that is configured on a PE router and an attachment circuit in a CE device.

To perform IPv4 cross connection between an AToM routed pseudowire and attachment circuit, use the following commands:

```
router(config)#l2vpn xconnect context xconnect-name
router(config-xconnect)#interworking ip
router(config-xconnect)#member pseudowire pw-number
router(config-xconnect)#member multilink <multilink-id>
```

To perform IPv6 cross connection between an AToM routed pseudowire and attachment circuit, use the following commands:

```
router(config)#l2vpn xconnect context xconnect-name
router(config-xconnect)#interworking ipv6
router(config-xconnect)#member pseudowire pw-number
router(config-xconnect)#member multilink <multilink-id>
```

The following example shows how to configure IPv6 interworking and cross connection under the attahment circuit for multilink interfaces:

```
l2vpn xconnect context temp
interworking ipv6
member Multilink1
member Pseudowire 1
```

Verifying L2VPN Interworking and Cross-Connect Configuration

You can verify L2VPN interworking configuration using the following **show** commands:

- **show mpls l2 vc <vc-id> detail**
- **show xconnect all**—Displays cross connect configuration.

- **show interfaces serial <interface-name>**—Provides interface details such as encapsulation, loopback, payload, MTU, bandwidth, and so on for a specific port.

```
router#show interfaces serial0/1/12.1
Serial0/1/12.1 is down, line protocol is down
  Hardware is unknown
  MTU 1500 bytes, BW 44210 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  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 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions alarm present
  DSU mode 0, bandwidth 0 Kbit, scramble 0, VC 3, non-inverted data
```

```
router#show xconnect all
Legend: XC ST=Xconnect State S1=Segment1 State S2=Segment2 State
UP=Up DN=Down AD=Admin Down IA=Inactive
SB=Standby HS=Hot Standby RV=Recovering NH=No Hardware

XC ST Segment 1 S1 Segment 2 S2
------+---------------------------------+--+---------------------------------+--
DN pri mpls 10.2.2.2:30 DN ac Mu30(PPP) UP
```

MLPPP Layer 3 Termination

Table 2: Feature History

Feature Name	Release Information	Description
MLPPP IP Termination on all Serial Physical and Logical Interfaces	Cisco IOS XE Bengaluru 17.5.1	This release supports Layer 3 termination using IPv6 addressing on MLPPP interfaces for the 1 port OC-48/STM-16 or 4 port OC-12/OC-3 / STM-1/STM-4 + 12 port T1/E1 + 4 port T3/E3 CEM interface module. In releases earlier, with IPv4 addressing, you can scale up to 512 MLPPP bundles. Now with IPv6 addressing, the MLPPP bundles can be scaled up to 1024.

Starting with Cisco IOS XE Bengaluru 17.5.1, you can perform Layer 3 termination using IPv6 addressing on MLPPP interfaces on Cisco RSP3 modules. In releases earlier, with IPv4 addressing, you can scale up to 512 MLPPP bundles. Now with IPv6 addressing, the MLPPP bundles can be scaled up to 1024.

Restrictions for MLPPP Layer 3 Termination

- In the Cisco IOS XE Bengaluru 17.5.1, the maximum number of Layer 3 MLPPP interfaces that are supported is 1000.
- MTU cannot be configured on the MLPPP bundle, and can be configured only on individual members.
- NxDS0 serial interfaces cannot be part of the MLPPP bundle.
- Starting with Cisco IOS XE Bengaluru 17.5.1, when you scale up MLPPP, ensure that you use the following commands under the interface:
  - **ppp timeout retry** <*retry no*>
  - **keepalive** <*keepalive no*>

How to Configure MLPPP Layer 3 Termination

Configuring MLPPP Layer 3 Termination

To configure MLPPP Layer 3 termination, use the following commands:

```
router#configure terminal
router(config)#interface Multilink multilinkno
router(config-if)#ip address ip-address subnetmask
router(config-if)#commit
```

The following example shows how to configure MLPPP Layer 3 termination for IPv4 address:

```
router#configure terminal
router#interface Multilink 1001
router#ip address 10.10.10.1 255.0.0.0
```

The following example shows how to configure MLPPP Layer 3 termination for IPv6 address:

```
router#configure terminal
router#interface Multilink 1001
router#ipv6 address 4001:1::2/64
```

MLPPP ACR Support for iMSG

Table 3: Feature History

Feature Name	Release Information	Description
MLPPP ACR support for IPv4 or IPv6 Interworking Multiservice Gateway (iMSG)	Cisco IOS XE Amsterdam 17.3.1	MLPPP ACR is supported for IPv4 or IPv6 iMSG on the Cisco RSP3 module. The restrictions for MLPPP interworking are applicable to iMSG ACR.

Starting with release Cisco IOS XE Cupertino 17.9.1, MLPPP bundle is supported on the RSP3 modules, only for TDM Interfaces.

For more information on supported modes and configuration, see iMSG ACR.

Restriction for MLPPP ACR

- The MLPPP ACR is supported only on the RSP3 3G module.
- Each router supports upto a total of 510 serial ACR interfaces. Hence, a total of 255 bundles can be configured, provided each MLPPP bundle has two links.
- Serial ACR and the other serial interface cannot be a part of the same bundle.
- Layer 3 termination not supported.
- A maximum number of links supported in an MLPPP bundle is 16.
- The change of encapsulation is not supported if the serial ACR interface is a part of the MLPPP bundle.

Note Restrictions for MLPPP interworking and iMSG ACR are applicable for MLPPP ACR.

Multilink PPP Minimum Links Mandatory

The Multilink PPP Minimum Links Mandatory feature enables you to configure the minimum number of links in a Multilink PPP (MLP) bundle required to keep that bundle active. You can enable this feature by entering the **ppp multilink min-links links mandatory** command. When you configure this command, all Network Control Protocols (NCPs) for an MLP bundle are disabled until the MLP bundle has the required minimum number of links. When a new link is added to the MLP bundle that brings the number of links up to the required minimum number of links, the NCPs are activated for the MLP bundle. When a link is removed from an MLP bundle, and the number of links falls below the required minimum number of links for that MLP bundle, the NCPs are disabled for that MLP bundle

How to Configure Multilink PPP Minimum Links Mandatory

Configuring Multilink PPP Minimum Links Mandatory

Before configuring multilink PPP minimum links mandatory, ensure that PPP and MLPPP are configured.

To configure the minimum number of links in an MLP bundle required to keep the bundle active, enter the following commands:

```
Router(config-if)#interface multilink multilink-id
Router(config-if)#ppp multilink links minimum <nooflinks> mandatory
```

The following example shows the multilink PPP minimum links mandatory configuration:

```
interface multilink 3
ppp multilink
ppp multilink min-links 5 mandatory
```

Verifying the Multilink PPP Minimum Links Mandatory Configuration

To verify the multilink PPP minimum links mandatory configuration, use the following **show** commands:

- **show running-config [interface type number] [linenum]**—Displays the contents of the currently running configuration file or the configuration for a specific interface, or map class information.
- **show interfaces multilink group-number**—Displays statistics for the specified multilink bundle.
- **show ppp multilink**—Displays information about all existing multilink bundles and their member links.
- **show interfaces multilink group-number stat**—Displays traffic statistics for a multilink bundle.

```
Router# show run int multilink 10
.
.
.Building configuration...

Current configuration : 160 bytes
!
interface Multilink10
no ip address
ppp multilink
```

```
ppp multilink links minimum 3 mandatory
ppp multilink group 10
ppp multilink endpoint string TEST
end
```

Multilink Frame Relay

Table 4: Feature History

Feature Name	Release Information	Description
Multilink Frame Relay (MFR) for IP Interworking	Cisco IOS XE Dublin 17.10.1	You can configure MFR encapsulation on serial interface for IPv4 and IPv6 interworking for the following interface modules: • One port OC-48/STM-16 or 4 port OC-12/OC-3 / STM-1/STM-4 + 12 port T1/E1 + 4 port T3/E3 CEM Interface Module (NCS4200-3GMS) • 1-Port OC-192 or 8-Port Low Rate CEM 20G Bandwidth Interface Module (NCS4200-1T8S-20CS) Multiple physical interfaces can be combined into a single bundle, and this frame relay interface supports more bandwidth than that is available from any single physical interface. The ease to add or remove physical interfaces dynamically so that you can modify the total bandwidth available on that interface. The resilience that is provided when multiple physical interfaces are provisioned on a single bundle so that when some of the physical interfaces fail, the bundle continues to support the frame relay service.

MFR for the User-to-Network Interface (UNI) and Network-to-Network Interface (NNI) provides physical interface emulation for frame relay devices. The emulated physical interface consists of one or more physical links that are aggregated into a single bundle of bandwidth. This service provides a frame-based inverse multiplexing function and solves most of the following problems on user-to-network in a frame relay network:

- Lack of required bandwidth availability due to facility constraints (for example, there is no E3/T3 service in a geographical region), or service offering restrictions (for example, there is no fractional E1/T1 service).
- The physical interface as an inflexible pool of bandwidth
- The physical interface as a single point of failure on the frame relay interface

MFR configuration is supported on the following controllers:

- T1 or E1
- T3 or E3
- SONET (CT-3 and VT-15 modes only)
- SDH (VC-1x mode only)

Limitations of IP Interworking on MFR Interfaces

- Sub interface with point-to-multipoint is not supported. Only point-to-point is supported.
- QoS, fragmentation, and interleaving are not supported.
- Sub interface number and DLCI number should be same for connect and cross connect scenarios.
- Local connect is supported with IETF only, not with CISCO (default).
- Local connect protection is not supported.
- Functional operations, for example shut or no shut, on sub interface are not supported.
- The **l2vpn** command-line interface (CLI) is not supported.
- MFR is not supported on the STS-1e mode.
- Changing configuration from FR to MFR is not supported. First, you should remove all the existing FR configurations and then reconfigure with MFR.

How to Configure IP Interworking on MFR Interface

Configuring MFR

To configure MFR, perform the following steps:

1. Configure a channel group.
2. Create an MFR interface.
3. Attach an MFR interface to a serial interface.
4. Configure an MFR sub interface.
5. Configure IPv4 or IPv6 profile (for example, serial_testipv4 or serial_testipv6).
6. Configure a cross-connection using sub interfaces.

The following example shows how to configure MFR.

In the example, interface MFR5 is created with the frame relay interface type as Data Communications Equipment (DCE) and Local Management Interface (LMI) type as Cisco. Once the MFR interface is created, this interface is attached to the serial interface 10/4/16.1. The MFR5 interface is attached to the serial interface SERIAL-ACR10.57. After this, the MFR sub interface MFR 5.100 is are created. Now, the pseudowire is created using the sub interface.

Configuring an MFR Interface and Attaching it to the Serial Interface, and and Configuring an MFR Sub Interface

```
router(config)#interface MFR 5
router(config-if)#no ip address
router(config-if)#frame-relay intf-type dce
router(config-if)#frame-relay lmi-type cisco
router(config-if)#end

router(config)#interface Serial0/4/16.1
router(config-if)#encapsulation frame-relay MFR5
router(config-if)#end

router(config)#interface MFR 5.100 point-to-point
router(config-if)#end
```

Configuring Local Connection using Sub Interface

IPv4

```
router(config)#connect mlfrsub21vlcv4 GigabitEthernet0/0/3 100 MFR1.100 100 interworking
ip
```

IPv6

```
router(config)#connect mlfrsub21vlcv6 GigabitEthernet0/0/3 150 MFR1.150 150 interworking
ipv6
```

Configuring a Cross-connection Using Sub Interfaces

IPv4

```
router(config)#connect mfr5_100 MFR5.100 100 l2transport
router(config-fr-pw-switching)#xconnect 3.3.3.3 100 encapsulation mpls pw-class
serial_testipv4
router(config-xconn)#end
```

IPv6

```
router(config)#connect mfr5_100 MFR5.100 100 l2transport
router(config-fr-pw-switching)#xconnect 3.3.3.3 100 encapsulation mpls pw-class
serial_testipv6
router(config-xconn)#end
```

You can verify the MFR configuration using the following **show xconnect all** command.

```
router#show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware
```

```
XC ST  Segment 1                          S1 Segment 2                           S2
------+---------------------------------+--+-----------------------------------+--
UP pri   ac MF5.100:100(FR DLCI)          UP mpls 3.3.3.3:100             UP
```

Configuring MFR ACR

You can configure MFR with Access Circuit Redundancy (ACR) only on the SONET (CT-3 and VT-15 modes).

To configure MFR ACR, perform the following steps:

1. Create an MFR interface.
2. Attaching an MFR interface to a serial interface
3. Create an MFR sub interface.
4. Configure IPv4 or IPv6 profile (for example, serial_testipv4 or serial_testipv6).
5. Configure a cross-connection using sub interfaces.

The following is a sample configuration for MLFR ACR.

In the example, two interfaces MFR4 and MFR5 are created with the frame relay interface type as Data Communications Equipment (DCE). Once the MFR interfaces are created, these interfaces are attached to the serial ACR interfaces. The MFR4 interface is attached to the serial interface SERIAL-ACR10.29 and the MFR5 interface is attached to the serial interface SERIAL-ACR10.57. Once these interfaces are attached, the MFR sub interfaces are created, for example, MFR4.101 and MFR5.151. Now, the pseudowires are created between the sub interfaces.

Creating an MFR interface

```
router(config)#interface MFR4
router(config-if)#no ip address
router(config-if)#frame-relay intf-type dce
router(config-if)#end
!

router(config)#interface MFR5
router(config-if)#no ip address
router(config-if)#frame-relay intf-type dce
router(config-if)#end
!
```

Attaching an MFR Interface to Serial Interface

```
router(config)#interface SERIAL-ACR10.29
router(config-if)#no ip address
router(config-if)#encapsulation frame-relay MFR4
router(config-if)#end
!
router(config)#interface SERIAL-ACR10.57
router(config-if)#no ip address
router(config-if)#encapsulation frame-relay MFR5
router(config-if)#end
!
```

Creating an MFR Sub Interface

```
router(config)#interface MFR4.101 point-to-point
```

```
router(config-if)#end

router(config)#interface MFR5.151 point-to-point
router(config-if)#end
!
```

Configuring a Cross-connection using Sub Interfaces

```
router(config)#connect mfr_101 MFR4.101 101 l2transport
router(config-fr-pw-switching)#xconnect 3.3.3.3 101 encapsulation mpls pw-class
serial_testipv4
router(config-xconnect)#end
!
router(config)#connect mfr_151 MFR5.151 151 l2transport
router(config-fr-pw-switching)#xconnect 3.3.3.3 151 encapsulation mpls pw-class
serial_testipv6
router(config-xconnect)#end
!
```

Verifying MFR Configuration

Use the following **show** commands to verify the MFR configuration.

- **show interface serial**
- **show xconnect all**
- **show ip interface brief**
- **show interfaces MFR*mfr-id***
- **show frame-relay multilink MFR*mfr-id***

```
router#show interfaces MFR1
MFR1 is up, line protocol is up
  Hardware is Multilink Frame Relay bundle interface
  MTU 1500 bytes, BW 18432 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 2 seconds on reset
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 4330, LMI stat sent  4330, LMI upd sent  0, DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:06, output never, output hang never
  Last clearing of "show interface" counters 12:03:41
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/120 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     108300 packets input, 1330362 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     108334 packets output, 1340642 bytes, 0 underruns
     Output 0 broadcasts (0 IP multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

```
     0 carrier transitions


router#show frame-relay multilink MFR1
Bundle: MFR1, State = up, class = A, fragmentation disabled
 BID = MFR1
 Bundle links:
  Serial0/4/7.12, HW state = up, link state = Up, LID = Serial0/4/7.12
  Serial0/4/7.11, HW state = up, link state = Up, LID = Serial0/4/7.11
  Serial0/4/7.10, HW state = up, link state = Up, LID = Serial0/4/7.10
  Serial0/4/7.9, HW state = up, link state = Up, LID = Serial0/4/7.9
  Serial0/4/7.8, HW state = up, link state = Up, LID = Serial0/4/7.8
  Serial0/4/7.7, HW state = up, link state = Up, LID = Serial0/4/7.7
  Serial0/4/7.6, HW state = up, link state = Up, LID = Serial0/4/7.6
  Serial0/4/7.5, HW state = up, link state = Up, LID = Serial0/4/7.5
  Serial0/4/7.4, HW state = up, link state = Up, LID = Serial0/4/7.4
  Serial0/4/7.3, HW state = up, link state = Up, LID = Serial0/4/7.3
  Serial0/4/7.2, HW state = up, link state = Up, LID = Serial0/4/7.2
  Serial0/4/7.1, HW state = up, link state = Up, LID = Serial0/4/7.1


Router#show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware

XC ST  Segment 1                         S1 Segment 2                          S2
------+--------------------------------+--+--------------------------------+--
UP pri   ac MF5.100:100(FR DLCI)         UP mpls 3.3.3.3:100             UP
```

Multilink Frame Relay Layer 3 Termination

Table 5: Feature History

Feature Name	Release Information	Description
Multilink Frame Relay (MFR) Layer 3 Termination	Cisco IOS XE Dublin 17.11.1	You can configure Layer 3 termination using IPv4 and IPv6 addressing on the MFR sub interfaces for the following interface modules: • NCS4200-3GMS and NCS4200-1T8S-20CS You can assign IP address on the MFR sub interface and terminate the Layer 3 traffic where ever required in the network.

On the RSP3 module, you can configure MFR interface on the following modes for the NCS4200-3GMS interface module:

- Electrical T1/E1
- Electrical T3/E3

- For Optical OCx/STM, mode—OCx-VT1.5, CT3, and AU4-VC11

On the RSP3 module, you can configure MFR interface on the following modes for the NCS4200-1T8S-20CS interface module:

- For Optical OCx/STM, mode—OCx-VT1.5, CT3, and AU4-VC11

Restrictions for MFR Layer 3 Termination

- QOS, fragmentation, and interleaving are not supported.
- Sub interface number and DLCI number should be same.
- OSPF and ISIS are only supported.
- On the RSP2 module, MFR IPv4 ping is not supported.

How to Configure MFR Layer 3 Termination

Configuring MFR Layer 3 Termination

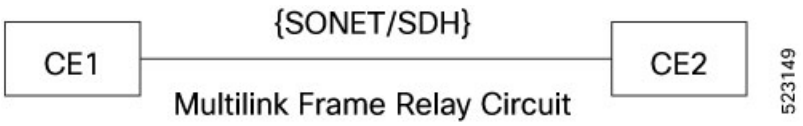
Scenario 1 - Configuring MFR Layer 3 Termination with IPv4 Address

Consider a scenario with the following configurations on CE1 and CE2. Both CE1 and CE2 are connected through serial interfaces and the traffic is carried through a multilink frame relay circuit.

CE1—On the serial interface 0/13/19.1 with frame relay as encapsulation, the default interface type as DTE, and default LMI type as Cisco, are configured. A sub interface MFR100.100 is created with DLCI as 100.

CE2—On the serial interface 0/13/19.1 with frame relay as encapsulation and interface type as DCE, a sub interface MFR100.100 is created with DLCI as 100.

Figure 1: Configuring MFR Layer 3 Termination with IPv4 Address



Configuring an MFR Interface and Sub Interface on CE1

```
CE1(config)#interface Serial0/13/19.1
CE1(config-if)#no ip address
CE1(config-if)#encapsulation frame-relay MFR100
CE1(config-if)#end


CE1(config-if)#interface MFR100
CE1(config-if)#no ip address
CE1(config-if)#frame-relay lmi-type ansi
CE1(config-if)#frame-relay intf-type dte
CE1(config-if)#end


CE1(config-if)#interface MFR100.100 point-to-point
```

```
CE1(config-subif)#ip address 192.10.2.1 255.255.255.0
CE1(config-subif)#frame-relay interface-dlci 100
CE1(config-fr-dlci)#end
```

Configuring an MFR Interface and Sub Interface on CE2

```
CE2(config)#interface Serial0/13/19.1
CE2(config-if)#no ip address
CE2(config-if)#encapsulation frame-relay MFR100
CE2(config-if)#end

CE2(config-if)#interface MFR100
CE2(config-if)#no ip address
CE2(config-if)#frame-relay lmi-type ansi
CE2(config-if)#frame-relay intf-type dce
CE2(config-if)#end

CE2(config-if)#interface MFR100.100 point-to-point
CE2(config-subif)#ip address 192.10.2.2 255.255.255.0
CE2(config-subif)#frame-relay interface-dlci 100
CE2(config-fr-dlci)#end
```

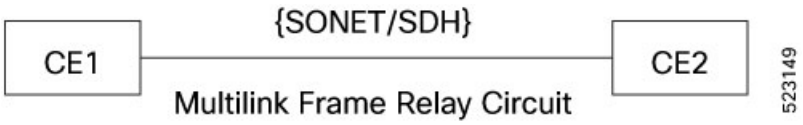
Scenario 2 - Configuring MFR Layer 3 Termination with IPv6 Address

Consider a scenario with the following configurations on CE1 and CE2. Both CE1 and CE2 are connected through serial interfaces and the traffic is carried through a multilink frame relay circuit.

CE1—An MFR interface MFR 100 is created with the interface type as DTE, and the default LMI type as ANSI. A sub interface MFR100.100 is created with DLCI as 100 and an IPv6 address as 2001:DB8:1::1.

CE2—An MFR interface MFR 100 is created with the interface type as DCE, and the default LMI type as ANSI. A sub interface MFR100.100 is created with DLCI as 100 and an IPv6 address as 2001:DB8:1::2.

Figure 2: Configuring MFR Layer 3 Termination with IPv6 Address



Configuring an MFR Interface and Sub Interface on CE1

```
CE1(config)#interface Serial0/13/19.1
CE1(config-if)#no ip address
CE1(config-if)#encapsulation frame-relay MFR100
CE1(config-if)#end

CE1(config-if)#interface MFR100
CE1(config-if)#no ip address
CE1(config-if)#frame-relay lmi-type ansi
CE1(config-if)#frame-relay intf-type dte
CE1(config-if)#end

CE1(config-if)#interface MFR100.100 point-to-point
CE1(config-subif)#ipv6 address 2001:10::1/64
CE1(config-subif)#ipv6 enable
CE1(config-subif)#frame-relay interface-dlci 100
CE1(config-fr-dlci)#end
```

```
CE1(config)#ipv6 unicast-routing
CE1(config)#end
```

Configuring an MFR Interface and Sub Interface on CE2

```
CE2(config)#interface Serial0/14/19.1
CE2(config-if)#no ip address
CE2(config-if)#encapsulation frame-relay MFR100
CE2(config-if)#end

CE2(config-if)#interface MFR100
CE2(config-if)#no ip address
CE2(config-if)#frame-relay lmi-type ansi
CE2(config-if)#frame-relay intf-type dce
CE2(config-if)#end

CE2(config-if)#interface MFR100.100 point-to-point
CE2(config-subif)#ipv6 address 2001:10::2/64
CE2(config-subif)#ipv6 enable
CE2(config-subif)#frame-relay interface-dlci 100
CE2(config-fr-dlci)#end

CE2(config)#ipv6 unicast-routing
CE2(config)#end
```

Verifying MFR Layer 3 Termination

Use the following **show** commands to verify the MFR Layer 3 termination:

- **show interface serial**
- **show ip interface brief**
- **show interfaces MFR1**
- **show frame-relay multilink MFR1**

```
CE1#show interfaces MFR1
MFR1 is up, line protocol is up
  Hardware is Multilink Frame Relay bundle interface
  MTU 1500 bytes, BW 18432 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 2 seconds on reset
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 4330, LMI stat sent  4330, LMI upd sent  0, DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:06, output never, output hang never
  Last clearing of "show interface" counters 12:03:41
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/120 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     108300 packets input, 1330362 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     108334 packets output, 1340642 bytes, 0 underruns
```

```
     Output 0 broadcasts (0 IP multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions

CE1#show frame-relay multilink MFR1
Bundle: MFR1, State = up, class = A, fragmentation disabled
 BID = MFR1
 Bundle links:
  Serial0/4/7.12, HW state = up, link state = Up, LID = Serial0/4/7.12
  Serial0/4/7.11, HW state = up, link state = Up, LID = Serial0/4/7.11
  Serial0/4/7.10, HW state = up, link state = Up, LID = Serial0/4/7.10
  Serial0/4/7.9, HW state = up, link state = Up, LID = Serial0/4/7.9
  Serial0/4/7.8, HW state = up, link state = Up, LID = Serial0/4/7.8
  Serial0/4/7.7, HW state = up, link state = Up, LID = Serial0/4/7.7
  Serial0/4/7.6, HW state = up, link state = Up, LID = Serial0/4/7.6
  Serial0/4/7.5, HW state = up, link state = Up, LID = Serial0/4/7.5
  Serial0/4/7.4, HW state = up, link state = Up, LID = Serial0/4/7.4
  Serial0/4/7.3, HW state = up, link state = Up, LID = Serial0/4/7.3
  Serial0/4/7.2, HW state = up, link state = Up, LID = Serial0/4/7.2
  Serial0/4/7.1, HW state = up, link state = Up, LID = Serial0/4/7.1
```

Multilink Frame Relay End-to-End Mode

Table 6: Feature History

Feature Name	Release Information	Description
Multilink Frame Relay (MFR) End-to-End Mode	Cisco IOS XE 17.13.1	This feature enables transporting legacy MFR traffic over packet switching core. You can now configure an end-to-end mode for frame relay pseudowire on the MFR interfaces for the following interface modules: • NCS4200-3GMS • NCS4200-1T8S-20CS

Multilink Frame Relay (MFR) is a packet transport protocol based on the adoption of the Frame Relay Forum's End-to-End MFR (FRF.15) and UNI/NNI MFR (FRF.16) Implementation Agreements. It provides physical interface emulation for frame relay devices by aggregating multiple physical circuits to transport data. However, the aggregation of physical circuits appears as one logical data pipe.

For more information on MFR, see Multilink Frame Relay, on page 11.

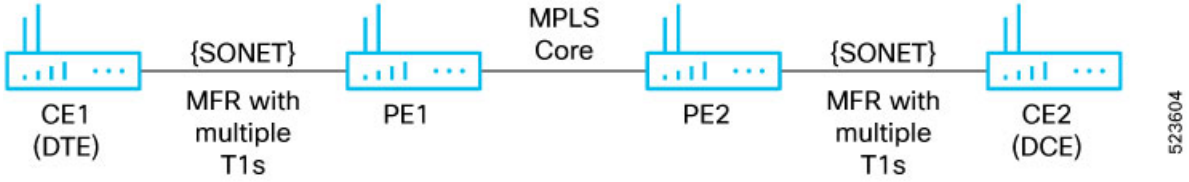
The MFR end-to-end mode feature specifies a method to aggregate multiple T1s into multiple FR bundles and to carry these bundles over an MPLS network. You can configure MFR end-to-end mode on SONET VT1.5 and CT-3 multiplexing modes for FR pseudowire.

How to Configure MFR End-to-End

The following section details how to configure MFR end-to-end.

Configuring MFR End-to-End

Consider a scenario where PE1 and PE2 are connected over MPLS core. CE1 is connected to PE1 and CE2 is connected to PE2.



CE1 Configuration

Follow the steps to configure CE1:

1. Configure MFR Interface.
2. Configure DLCI on Member Serial Interface.

MFR Interface Configuration

```
interface MFR20
 ip address 192.0.2.1 255.255.255.0
 frame-relay end-to-end
 frame-relay map ip 192.0.2.2 100
```

Note MFR ID should be the same for PE1 and CE1.

DLCI Configuration

```
interface Serial0/3/16.1
 no ip address
 encapsulation frame-relay MFR20
 frame-relay interface-dlci 101

interface Serial0/3/16.2
 no ip address
 encapsulation frame-relay MFR20
 frame-relay interface-dlci 102
```

PE1 Configuration

Follow the steps to configure PE1:

1. Configure MFR interface.
2. Configure DLCI on mMember serial interface.
3. Configure xconnect on MFR interface.

MFR Interface Configuration

```
interface MFR20
 no ip address
```

```
 frame-relay end-to-end
 frame-relay intf-type dce
```

DLCI Configuration

```
interface Serial0/7/0.1
 no ip address
 encapsulation frame-relay MFR20
 frame-relay interface-dlci 101

interface Serial0/7/0.2
 no ip address
 encapsulation frame-relay MFR20
 frame-relay interface-dlci 102
```

XConnect Configuration

```
pseudowire-class MFR_E2E
 encapsulation mpls
 interworking ip
 no control-word
 status
 status control-plane route-watch
 switching tlv


connect mfre2e MFR20 100 l2transport
 xconnect 198.51.100.1 200 encapsulation mpls pw-class MFR_E2E
```

Verify PE1 Configuration

```
PE1#show run | sec MFR_E2E
pseudowire-class MFR_E2E
 encapsulation mpls
 interworking ip
 no control-word
 status
 status control-plane route-watch
 switching tlv


PE1#sh run | sec connect
connect mfre2e MFR20 100 l2transport
 xconnect 203.0.113.1 200 encapsulation mpls pw-class MFR_E2E
```

PE2 Configuration

Follow the steps to configure PE2:

1. Configure MFR Interface.
2. Configure DLCI on member serial interface.
3. Configure xconnect on MFR interface.

MFR Interface Configuration

```
interface MFR30
 no ip address
 frame-relay end-to-end
 frame-relay intf-type dce
```

DLCI Configuration

```
interface Serial0/7/0.1
 no ip address
 encapsulation frame-relay MFR30
 frame-relay interface-dlci 101


interface Serial0/7/0.2
 no ip address
 encapsulation frame-relay MFR30
 frame-relay interface-dlci 102
```



Note MFR ID should be the same for PE2 and CE2.

XConnect Configuration

```
pseudowire-class MFR_E2E
 encapsulation mpls
 interworking ip
 no control-word
 status
 status control-plane route-watch
 switching tlv


connect mfre2e MFR30 100 l2transport
 xconnect 203.0.113.1 200 encapsulation mpls pw-class MFR_E2E
```

Verify PE2 Configuration

```
PE2#sh run | sec MFR_E2E
pseudowire-class MFR_E2E
 encapsulation mpls
 interworking ip
 no control-word
 status
 status control-plane route-watch
 switching tlv


PE2#sh run | sec connect
connect mfre2e MFR30 100 l2transport
 xconnect 203.0.113.1 200 encapsulation mpls pw-class MFR_E2E
```

CE2

Follow the steps to configure CE2:

1. Configure MFR interface.
2. Configure DLCI on member serial interface.

MFR Interface Configuration

```
interface MFR30
 ip address 192.0.2.2 255.255.255.0
 frame-relay end-to-end
 frame-relay map ip 192.0.2.1 100
```

DLCI Configuration

```
interface Serial0/3/16.1
 no ip address
 encapsulation frame-relay MFR30
 frame-relay interface-dlci 101


interface Serial0/3/16.2
 no ip address
 encapsulation frame-relay MFR30
 frame-relay interface-dlci 102
```

Verifying MFR End-to-End

Use the following **show** commands to verify the MFR end-to-end configuration:

- **show interface *interface-name***
- **show xconnect all**
- **show mpls l2transport vc 200 detail | in MPLS**

You can verify details about the MFR interface and traffic using the **show interface *interface-name***.

```
PE1#show interface MFR20
MFR20 is up, line protocol is up
  Hardware is Multilink Frame Relay bundle interface
  MTU 1500 bytes, BW 21504 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 2 seconds on reset
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 1719, LMI stat sent  1719, LMI upd sent  0, DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:04, output never, output hang never
  Last clearing of "show interface" counters 04:50:59
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 1
  Queueing strategy: fifo
  Output queue: 0/120 (size/max)
5 minute input rate 9000 bits/sec, 9 packets/sec
     5 minute output rate 9000 bits/sec, 9 packets/sec
        102901 packets input, 11762696 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     102587 packets output, 11726574 bytes, 0 underruns
     Output 0 broadcasts (0 IP multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

You can verify the MFR xconnect circuit status using the **show xconnect all**.

```
PE1#show xconnect all
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                         S2
```

```
------+---------------------------------+--+---------------------------------+--
UP pri   ac MF20:100(FR DLCI)            UP mpls 3.3.3.3:200                  UP
```

You can verify and check the MPLS virtual circuit label information using the **show mpls l2transport vc 200 detail | in MPLS** command.

```
PE1#show mpls l2transport vc 200 detail | in MPLS
    MPLS VC labels: local 17, remote 17
```