Multilink Frame Relay over L2TPv3/AToM

First Published: May 6, 2004
Last Updated: November 17, 2006

This feature enables Multilink Frame Relay switching over Layer 2 Tunnel Protocol Version 3 (L2TPv3) and Any Transport over MPLS (AToM). The feature works with like-to-like interfaces and disparate interfaces (L2VPN interworking).

Multilink Frame Relay is the logical grouping of one or more physical interfaces between two devices of the User-to-Network Interface/Network-to-Network Interface (UNI/NNI) as one single Frame Relay data link.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Multilink Frame Relay over L2TPv3/AToM" section on page 22.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring Multilink Frame Relay over L2TPv3/AToM

Before configuring Multilink Frame Relay over L2TPv3/AToM, you should understand how to configure Layer 2 virtual private networks (VPNs) and Multilink Frame Relay. See the "Additional References" section on page 16 for pointers to the feature modules that explain how to configure and use those features.

Restrictions for Configuring Multilink Frame Relay over L2TPv3/AToM

- Only data-link connection identifier (DLCI)-to-DLCI switching, where each DLCI maps to its own pseudowire, is supported. Port-port mode (also known as HDLC mode), where the entire content of the port, including the Local Management Interface (LMI), is carried across a single pseudowire, is not supported.
- The following functionality is not supported:
  - UNI/NNI or end-to-end fragmentation
  - Nonstop forwarding/stateful switchover
  - Four-byte DLCIs
- On the Cisco 7500 series routers, all bundle links must reside on the same port adapter (PA) of the Versatile Interface Processor (VIP). Links spreading across PAs are not supported.
- Cisco 7500 series routers support the VIP6-80, VIP4-80, VIP4-50, VIP2-50, CH-STM1, CT3/CE3, CT1/CE1, PA-4T+, and PA-8T port adapters.
- On the Cisco 12000 series routers, Multilink Frame Relay is supported only on the following pluggable modules: Cisco 4-port channelized T3 (DSO) shared port adapter, Cisco 8-port channelized T1/E1 shared port adapter, and the Cisco 1-port channelize OC-3/STM-1shared port adapter.

Information About Configuring Multilink Frame Relay over L2TPv3/AToM

To configure Multilink Frame Relay over L2TPv3/AToM, you need to understand the following concepts:

- Multilink Frame Relay over L2TPv3/AToM, page 2
- Internetworking Support for Multilink Frame Relay, page 3
- Quality of Service Support for Multilink Frame Relay over L2TPv3/AToM, page 3

Multilink Frame Relay over L2TPv3/AToM

Multilink Frame Relay over L2TPv3/AToM supports the following functionality:

- Permanent virtual circuit (PVC) status signaling
- LMI types cisco, q933a, and ANSI
- Sequencing
- Frame Relay policing (nondistributed)
- Type of service (ToS) marking for L2TPv3

Internetworking Support for Multilink Frame Relay

Interworking support for Multilink Frame Relay interfaces supports the following functionality:

- Frame Relay to Ethernet/VLAN (Ethernet and IP interworking)
- Frame Relay to PPP and ATM (IP interworking)
- Cisco and Internet Engineering Task Force (IETF) encapsulation on the customer-edge (CE) router
- Sequencing
- LMI interworking to notify CE routers of PVC status changes

Quality of Service Support for Multilink Frame Relay over L2TPv3/AToM

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

L2VPN quality of service (QoS) features supported for Frame Relay are also supported with the Multilink Frame Relay over L2TPv3/AToM feature. You can attach an input service policy to the Multilink Frame Relay interface or individual DLCIs on the interface using the map-class mechanism to police or mark the traffic. You can attach an output policy to the Multilink Frame Relay (MFR) interface to perform class-based queueing, including per-DLCI queueing using the **match fr-dlci** command.

The following ingress QoS features are supported with the Multilink Frame Relay over L2TPv3/AToM feature:

- Interface input policy matching on the discard eligibility (DE) bit to set Multiprotocol Label Switching (MPLS) EXP or tunnel differentiated services code point (DSCP).
- Virtual circuit (VC) input policy configured with a color-aware, two-rate, three-color policer using the DE bit as input color and setting the MPLS EXP bit or tunnel DSCP bit based on color.

Note You cannot use the VC-level and interface-level input policies at the same time on the same interface.

The following egress QoS features are supported with the Multilink Frame Relay over L2TPv3/AToM feature:

- Egress queueing using tail drop or discard class-based weighted random early detection (WRED). You can use the latter with a core interface input policy to set the discard class based on the MPLS EXP or tunnel DSCP.
- Interface output policy matching on QoS group (selected by MPLS EXP or tunnel DSCP).
- Interface aggregate shaping policy with queueing policy.
- VC output shaping policy with tail drop or discard class-based WRED.

- Forward explicit congestion notification (FECN)/backward explicit congestion notification (BECN) marking.

Note You cannot use VC-level and interface-level output policies at the same time on the same interface.

Note Egress queueing and shaping policies are not supported with Multilink Frame Relay on the Cisco 7200 series routers.

How to Configure Multilink Frame Relay over L2TPv3/AToM

This section contains the following procedures:

- Configuring a Multilink Frame Relay Bundle Interface (required)
- Configuring a Multilink Frame Relay Bundle Link Interface (required)
- Connecting Frame Relay PVCs Between Routers (required)
- Verifying Multilink Frame Relay over L2TPv3/AToM, page 7 (optional)

Configuring a Multilink Frame Relay Bundle Interface

Configure a bundle interface to aggregate bandwidth of multiple member links under a single interface to one virtual pipe. To configure a bundle interface for Multilink Frame Relay, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface mfr** *number*
4. **frame-relay multilink bid** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface mfr** *number* **Example:** `Router(config)# interface mfr 1`	Configures a multilink Frame Relay bundle interface and enters interface configuration mode.
Step 4	**frame-relay multilink bid** *name* **Example:** `Router(config-if)# frame-relay multilink bid int1`	(Optional) Assigns a bundle identification name to a multilink Frame Relay bundle. **Note** The bundle identification (BID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.

Configuring a Multilink Frame Relay Bundle Link Interface

Configuring a Multilink Frame Relay bundle link interface allows you to combine bandwidth of multiple lower-speed serial links into a single large pipe and avoid the need of upgrading or purchasing new hardware. To configure a bundle link interface for Multilink Frame Relay, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *number*
4. **encapsulation frame-relay mfr** *number* [*name*]
5. **frame-relay multilink lid** *name*
6. **frame-relay multilink hello** *seconds*
7. **frame-relay multilink ack** *seconds*
8. **frame-relay multilink retry** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface serial** *number* **Example:** `Router(config)# interface serial 1/1`	Configures an interface and enters interface configuration mode.
Step 4	**encapsulation frame-relay mfr** *number* [*name*] **Example:** `Router(config-if)# encapsulation frame-relay mfr 1`	Creates a multilink Frame Relay bundle link and associates the link with a bundle. **Tips** To minimize latency that results from the arrival order of packets, we recommend bundling physical links of the same line speed in one bundle.
Step 5	**frame-relay multilink lid** *name* **Example:** `Router(config-if)# frame-relay multilink lid four`	(Optional) Assigns a bundle link identification name with a multilink Frame Relay bundle link. **Note** The bundle link identification (LID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.
Step 6	**frame-relay multilink hello** *seconds* **Example:** `Router(config-if)# frame-relay multilink hello 20`	(Optional) Configures the interval at which a bundle link will send out hello messages. The default value is 10 seconds.
Step 7	**frame-relay multilink ack** *seconds* **Example:** `Router(config-if)# frame-relay multilink ack 10`	(Optional) Configures the number of seconds that a bundle link will wait for a hello message acknowledgment before resending the hello message. The default value is 4 seconds.
Step 8	**frame-relay multilink retry** *number* **Example:** `Router(config-if)# frame-relay multilink retry 5`	(Optional) Configures the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment. The default value is 2 tries.

Connecting Frame Relay PVCs Between Routers

By connecting Frame Relay PVCs between routers, you can integrate Frame Relay over a Level 2 VPN backbone, which allows you to use your existing Frame Relay network without upgrading. To connect Frame Relay PVCs between routers, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **connect** *connection-name* **mfr** *number dlci* **l2transport**
4. **xconnect** *peer-router-id vcid* **encapsulation mpls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**connect** *connection-name* **mfr** *number dlci* **l2transport** **Example:** `Router(config)# connect fr1 mfr 1 100 l2transport`	Defines connections between Frame Relay PVCs. • Using the **l2transport** keyword specifies that the PVC will not be a locally switched PVC, but will be tunneled over the backbone network. • The *connection-name* argument is a text string that you provide. • The *dlci* argument is the DLCI number of the PVC that will be connected. Enters connect configuration submode.
Step 4	**xconnect** *peer-router-id vcid* **encapsulation mpls** **Example:** `Router(config-fr-pw-switching)# xconnect 10.0.0.1 123 encapsulation mpls`	Creates the VC to transport the Layer 2 packets. In a DLCI-to-DLCI connection type, Frame Relay over MPLS uses the **xconnect** command in connect configuration submode.

Verifying Multilink Frame Relay over L2TPv3/AToM

To verify the configuration of Multilink Frame Relay, perform the following steps. The tunnel and session should be in the established (est) state.

SUMMARY STEPS

1. **show l2tunnel**
2. **show mpls forwarding**

DETAILED STEPS

Step 1 **show l2tunnel**

On both PE routers, use the following command to verify the configuration of Multilink Frame Relay over L2TPv3:

```
PE1# show l2tunnel
```

```
 Tunnel and Session Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TPclass
35788 41451 FRWI1         est    10.9.9.9         0     1        l2tp_default_cl

LocID      RemID      TunID      Username, Intf/         State
                                 Vcid, Circuit
8161       54072      35788      6, MF1:206              est

PE2# show l2tunnel

 Tunnel and Session Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TPclass
41451 35788 FRWI3         est    10.8.8.8         0     1

LocID      RemID      TunID      Username, Intf/          State
                                 Vcid, Circuit
54072      8161       41451      6, Fa0/1.6:6             est
```

Step 2 **show mpls forwarding**

On both PE routers, use the following command to verify the configuration of Multilink Frame Relay over MPLS:

```
PE1# show mpls forwarding

Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     10.0.0.0/24      0          PO4/1/0    point2point
17     Untagged    l2ckt(5)         0          MF1        point2point
18     Untagged    l2ckt(6)         0          MF1        point2point
19     17          10.9.9.9/32      0          PO4/1/0    point2point

PE2# show mpls forwarding

Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     16          10.8.8.8/32      0          PO2/0      point2point
17     Pop tag     10.13.0.0/24     0          PO2/0      point2point
18     Untagged    l2ckt(5)         2244       MF2        point2point
19     Untagged    l2ckt(6)         510        MF2        point2point
```

Configuration Examples for Multilink Frame Relay over L2TPv3/AToM

This section includes the following configuration examples:

Frame Relay-to-Frame Relay over L2TPv3 on Multilink Frame Relay Interfaces: Example

The following example sets up Multilink Frame Relay interfaces to transport Frame Relay data between PE routers:

PE1

```
configure terminal
ip cef distributed
frame-relay switching
!
interface loopback 0
 ip address 10.8.8.8 255.255.255.255
 no shutdown
!
pseudowire-class fr-xconnect
 encapsulation l2tp
 protocol l2tpv3
 ip local interface loopback0
!
controller T3 1/1/1
t1 1 framing esf
t1 1 clock source internal
t1 1 channel-group 1 timeslots 1-24 speed 64
!
t1 2 framing esf
t1 2 clock source inter
t1 2 channel-group 1 timeslots 1-24 speed 64
!
interface mfr 1
 encapsulation frame-relay
 logging event dlci-status-change
 frame-relay intf-type nni
 no shutdown
!
interface Serial1/1/1/1:1
  encapsulation frame-relay mfr1
interface Serial1/1/1/2:1
  encapsulation frame-relay mfr1
!
interface POS4/1/0
 clock source internal
 ip address 10.13.0.0 255.255.255.0
 no shutdown
 no fair-queue
!
connect fr-fr mfr1 206 l2
 xconnect 10.9.9.9 6 pw-class fr-xconnect
!
router ospf 10
 network 10.13.0.0 0.0.0.0 area 0
 network 10.8.8.8 0.0.0.0 area 0
end
```

PE2

```
configure terminal
ip routing
ip cef
frame-relay switching
!
interface loopback 0
 ip address 10.9.9.9 255.255.255.255
 no shutdown
!
interface p2/0
 clock source internal
 ip address 10.14.0.2 255.255.255.0
 no shutdown
 no fair-queue
!
controller T3 3/1
t1 1 framing esf
t1 1 clock source internal
t1 1 channel-group 1 timeslots 1-24 speed 64
!
t1 2 framing esf
t1 2 clock source internal
t1 2 channel-group 1 timeslots 1-24 speed 64
!
interface mfr2
 encapsulation frame-relay
 logging event dlci-status-change
 frame-relay intf-type dce
 no shutdown
!
interface serial3/1/1:1
 encapsulation frame-relay mfr2
!
interface s3/1/2:1
 encapsulation frame-relay mfr2
!
pseudowire-class fr-xconnect
 encapsulation l2tpv3
 protocol l2tpv3
 ip local interface loopback0
!
connect fr-fr mfr2 306 l2transport
 xconnect 10.8.8.8 6 pw-class fr-xconnect
!
router ospf 10
 network 10.14.0.2 0.0.0.0 area 0
 network 10.9.9.9 0.0.0.0 area 0
end
```

Frame Relay-to-Ethernet VLAN Interworking over L2TPv3 on Multilink Frame Relay Interfaces: Example

The following example sets up Multilink Frame Relay interfaces to perform Frame Relay-to-Ethernet VLAN interworking between PE routers. The example uses IP interworking, also referred to as routed interworking.

PE1

```
configure terminal
ip cef distributed
frame-relay switching
!
!
interface loopback 0
     ip address 10.8.8.8 255.255.255.255
     no shutdown
!
pseudowire-class ip
     encapsulation l2tp
     interworking ip
     ip local interface loopback0
!
interface mfr 1
     encapsulation frame-relay
     logging event dlci-status-change
     no shutdown
     frame-relay intf-type nni
!
interface Serial1/1/1/1:1
      encapsulation frame-relay mfr1
interface Serial1/1/1/2:1
      encapsulation frame-relay mfr1
!
interface POS4/1/0
     clock source internal
     ip address 13.0.0.2 255.255.255.0
     no shutdown
     no fair-queue
!
connect fr-vlan mfr1 206 l2
     xconnect 9.9.9.913.0.0.2 6 pw-class ip
!
router ospf 10
     network 10.13.0.2 0.0.0.0 area 0
     network 10.8.8.8 0.0.0.0 area 0
end
```

PE2

```
configure terminal
ip routing
ip cef
frame-relay switching
!
interface loopback 0
     ip address 10.9.9.9 255.255.255.255
     no shutdown
!
pseudowire-class ip
     encapsulation l2tp
     interworking ip
     ip local interface loopback0
!
interface p2/0
     clock source internal
     ip address 10.14.0.2 255.255.255.0
     no shutdown
     no fair-queue
!
interface FastEthernet0/1
      no shutdown
!
interface FastEthernet0/1.6
     encapsulation dot1Q 6
     xconnect 10.8.8.8 6 pw-class ip
     no shutdown
!
router ospf 10
     network 10.14.0.2 0.0.0.0 area 0
     network 10.9.9.9 0.0.0.0 area 0
!
end
```

Frame Relay-to-Ethernet Interworking over MPLS on Multilink Frame Relay Interfaces: Example

The following example sets up Multilink Frame Relay interfaces to perform Frame Relay-to-Ethernet interworking between PE routers. The example uses IP interworking, also referred to as routed interworking.

PE1

```
configure terminal
ip cef distributed
frame-relay switching
!
!
interface loopback 0
     ip address 10.8.8.8 255.255.255.255
     no shutdown
!
interface mfr 1
     encapsulation frame-relay
     logging event dlci-status-change
     no shutdown
     frame-relay intf-type nni
!
interface Serial1/1/1/1:1
      encapsulation frame-relay mfr1
interface Serial1/1/1/2:1
      encapsulation frame-relay mfr2
!
interface POS4/1/0
     clock source internal
     ip address 10.13.0.2 255.255.255.0
     no shutdown
     mpls ip
!
router ospf 10
     network 10.13.0.2 0.0.0.0 area 0
     network 10.8.8.8 0.0.0.0 area 0
!
mpls label protocol ldp
mpls ldp router-id loopback0
mpls ip
!
pseudowire-class atom
     encapsulation mpls
     interworking ip
!
connect fr-eth mfr1 207 l2
     xconnect 10.9.9.9 7 pw-class atom
!
end
```

PE2

```
configure terminal
ip routing
ip cef
frame-relay switching
!
interface loopback 0
     ip address 10.9.9.9 255.255.255.255
     no shutdown
!
interface POS2/0
     clock source internal
     ip address 10.14.0.2 255.255.255.0
     no shutdown
     no fair-queue
     mpls ip
!
router ospf 10
     network 10.14.0.2 0.0.0.0 area 0
     network 10.9.9.9 0.0.0.0 area 0
!
mpls label protocol ldp
mpls ldp router-id loopback0
mpls ip
!
pseudowire-class atom
     encapsulation mpls
     interworking ip
!
interface FastEthernet0/1
     xconnect 10.8.8.8 7 pw-class atom
     no shutdown
!
end
```

MQC Color-Aware Policing: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example configures a VC input policy with a color-aware, two-rate, three-color policing method using a DE bit as input color and setting the tunnel Differentiated Services Code Point (DSCP) based on color. Packets in excess of peak rates are discarded.

```
class-map not-fr-de
    match not fr-de
!
policy-map police
    class class-default
    police cir 64000 pir 256000
    conform-color not-fr-de
    conform-action set-dscp-tunnel-transmit  af31
    exceed-action set-dscp-tunnel-transmit af32
    violate-action drop
!
interface MFR1
    frame-relay interface-dlci 206 switched
    class police
!
connect fr-vlan mfr1 206 l2
    xconnect 10.9.9.9 6 pw-class ip
!
map-class frame-relay police
    service-policy input police
```

DE Bit Matching: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example shows the configuration of an interface input policy matching on the DE bit to set the tunnel DSCP:

```
class-map de
     match fr-de
!
policy-map de
     class de
          set ip dscp tunnel af32
 class class-default
      set ip dscp tunnel af31
!
interface MFR1
     service-policy input de
```

DLCI-Based queueing: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example shows the configuration of an interface output policy matching on a QoS group based on the DLCI:

```
class-map dlci100
     match fr-dlci 100
class-map dlci200
     match fr-dlci 200
!
policy-map dlci
     class dlci100
          bandwidth percent 10
     class dlci200
          bandwidth percent 20
!
interface MFR1
     service-policy output dlci
```

Discard Class-Based WRED: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example shows the configuration of an interface output policy matching on a QoS group based on the tunnel DSCP:

```
class-map conform
     match ip dscp af31
     match mpls experimental 4
class-map exceed
     match ip dscp af32
     match mpls experimental 3
class-map cos1
     match qos-group 1
!
policy-map core
     class conform
          set qos-group 1
          set discard-class 1
     class exceed
          set qos-group 1
          set discard-class 2
!
policy-map wred
     class cos1
          bandwidth percent 40
          random-detect discard-class-based
          random-detect discard-class 1 20 30 10
          random-detect discard-class 2 1 9 10
!
interface POS1/0
     service-policy input core
!
interface MFR1
     service-policy output wred
```

Aggregate Shaping: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example shows the configuration of an interface aggregate shaping policy with a DLCI-based queueing policy:

```
class-map dlci205
    match fr-dlci 205
class-map dlci206
    match fr-dlci 206
!
policy-map dlci
    class dlci205
        bandwidth 128
    class dlci206
        bandwidth 256
!
policy-map shape
    class class-default
        shape average 512000 2048 2048
        service-policy dlci
!
interface MFR1
    service-policy output shape
```

VC Shaping: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example shows the configuration of a VC output shaping policy with discard class-based WRED:

```
class-map conform
    match mpls experimental 4
class-map exceed
    match mpls experimental 3
class-map cos1
    match qos-group 1
!
policy-map core
    class conform
        set qos-group 1
        set discard-class 1
    class exceed
        set qos-group 1
        set discard-class 2
!
policy-map vc-wred
    class class-default
        bandwidth percent 40
        random-detect discard-class-based
        random-detect discard-class 1 20 30 10
        random-detect discard-class 2 1 9 10
!
policy-map shape
```

```
    class class-default
        shape average 512000 2048 2048
        service-policy vc-wred
!
interface POS4/1/0
    service-policy input core
!
interface MFR1
    frame-relay interface-dlci 206 switched
        class shape
!
map-class frame-relay shape
    service-policy output shape
```

FECN/BECN Marking: Example

Note Quality of Service features are not supported in Cisco IOS Release 12.4(11)T.

The following example shows the configuration of an output policy that configures BECN and FECN bits:

```
policy-map dlci
     class dlci100
          bandwidth percent 10
     class dlci200
          bandwidth percent 20
      set fr-fecn-becn 1
interface MFR1
     service-policy output dlci
     frame-relay congestion-management
          threshold ecn 20
```

Additional References

The following sections provide references related to the Multilink Frame Relay over L2TPv3/AToM feature.

Related Documents

Related Topic	Document Title
Multilink Frame Relay	• For the Cisco 7500 series routers: *Distributed Multilink Frame Relay (FRF.16)* • For the Cisco 7200 series routers: *Multilink Frame Relay (FRF.16)*
L2VPN interworking	L2VPN Interworking
Layer 2 Tunneling Protocol, Version 3	*L2TPV3*
Layer 2 local switching	*Layer 2 Local Switching*

Standards

Standard	Title
draft-martini-l2circuit-trans-mpls-08.txt	*Transport of Layer 2 Frames Over MPLS*
draft-martini-l2circuit-encap-mpls-04.txt	*Encapsulation Methods for Transport of Layer 2 Frames Over MPLS*
draft-ietf-l2tpext-l2tp-base-03.txt	*Layer Two Tunneling Protocol (Version 3)*

MIBs

MIB	MIBs Link
• Cisco Frame Relay MIB (CISCO-FRAME-RELAY-MIB.my) • Interfaces MIB (IF-MIB.my) • MPLS LDP MIB (MPLS-LDP-MIB.my)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2661	*Layer Two Tunneling Protocol*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport

Command Reference

This section documents modified commands only.

- **xconnect**

xconnect

To bind an attachment circuit to a pseudowire, use the **xconnect** command in one of the supported configuration modes. To restore the default values, use the **no** form of this command.

xconnect *peer-ip-address vcid pseudowire-parameters* [**sequencing** {**transmit** | **receive** | **both**}]

no xconnect

Syntax Description

peer-ip-address	IP address of the remote provider edge (PE) peer.
vcid	The 32-bit identifier of the virtual circuit between the PE routers.
pseudowire-parameters	Encapsulation and pseudowire class parameters to be used for the attachment circuit. At least one of the following pseudowire parameters must be configured: • **encapsulation** {**l2tpv3** [**manual**] | **mpls**}— Specifies the tunneling method to encapsulate the data in the pseudowire: – **l2tpv3**—Specifies L2TPv3 as the tunneling method. – **manual**—Specifies that no signaling is to be used in the attachment circuit. This keyword places the router in xconnect configuration mode for manual configuration the attachment circuit. – **mpls**—Specifies Multiprotocol Label Switching (MPLS) as the tunneling method. • **pw-class** *pw-class-name*—Specifies the pseudowire class configuration from which the data encapsulation type is taken. This option is mandatory if you select an encapsulation method.
sequencing	(Optional) Sets the sequencing method to be used for packets received or sent.
transmit	Sequences data packets received from the attachment circuit.
receive	Sequences data packets sent into the attachment circuit.
both	Sequences data packets that are both sent and received from the attachment circuit.

Command Default

The attachment circuit is not bound to the psuedowire.

Command Modes

Interface configuration
l2transport configuration (for ATM)
connect configuration mode

Command History

Release	Modification
12.0(23)S	This command was introduced.
12.0(28)S	Support was added for Multilink Frame Relay connections.

Release	Modification
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
12.4(11)T	This command was integrated into Cisco IOS Release 12.4(11)T.

Usage Guidelines

The combination of the *peer-ip-address* and *vcid* arguments must be unique on the router. Each xconnect configuration must have a unique combination of *peer-ip-address* and *vcid* configuration.

Note If the remote router is a Cisco 12000 series Internet router, the *peer-ip-address* argument must specify a loopback address on that router.

The same *vcid* value that identifies the attachment circuit must be configured using the **xconnect** command on the local and remote provider edge (PE) router. The vcid creates the binding between a pseudowire and an attachment circuit.

For L2TPv3, to manually configure the settings used in the attachment circuit, use the **manual** keyword in the **xconnect** command. This configuration is called a static session. The router is placed in xconnect configuration mode, and you can then configure the following options:

- Local and remote session identifiers (using the **l2tp id** command) for local and remote PE routers at each end of the session.
- Size of the cookie field used in the L2TPv3 headers of incoming (sent) packets from the remote PE peer router (using the **l2tp cookie local** command).
- Size of the cookie field used in the L2TPv3 headers of outgoing (received) L2TP data packets (using the **l2tp cookie remote** command).
- Interval used between sending hello keepalive messages (using the **l2tp hello** command).

For L2TPv3, if you do not enter **encapsulation l2tpv3 manual** in the **xconnect** command, the data encapsulation type for the L2TPv3 session is taken from the encapsulation type configured for the pseudowire class specified with the **pseudowire-class** *pw-class-name* command.

The **pw-class** *pw-class-name* value binds the xconnect configuration of an attachment circuit to a specific pseudowire class. In this way, the pseudowire class configuration serves as a template that contains settings used by all attachment circuits bound to it with the **xconnect** command.

Note If you specify the **encapsulation** keywords, you must specify the **pw-class** keyword.

Examples

The following example configures xconnect service for an Ethernet interface by binding the Ethernet circuit to the pseudowire named 123 with a remote peer 10.0.3.201. The configuration settings in the pseudowire class named vlan-xconnect are used.

```
Router(config)# interface Ethernet0/0.1
Router(config-if)# xconnect 10.0.3.201 123 pw-class vlan-xconnect
```

The following example enters xconnect configuration mode and manually configures L2TPv3 parameters for the attachment circuit:

```
Router(config)# interface Ethernet 0/0
```

```
Router(config-if)# xconnect 10.0.3.201 123 encapsulation l2tpv3 manual pw-class ether-pw
Router(config-if-xconn) l2tp id 222 111
Router(config-if-xconn) l2tp cookie local 4 54321
Router(config-if-xconn) l2tp cookie remote 4 12345
Router(config-if-xconn) l2tp hello l2tp-defaults
```

Related Commands

Command	Description
show xconnect	Displays information about xconnect attachment circuits and pseudowires
pseudowire-class	Configures a template of pseudowire configuration settings used by the attachment circuits transported over a pseudowire.
l2tp-class	Configures a template of L2TP control plane configuration settings that can be inherited by different pseudowire classes.
l2tp cookie local	Configures the size of the cookie field used in the L2TPv3 headers of incoming packets received from the remote PE peer router.
l2tp cookie remote	Configures the size of the cookie field used in the L2TPv3 headers of outgoing packets sent from the local PE peer router.
l2tp hello	Specifies the use of a hello keepalive setting contained in a specified L2TP class configuration for a static L2TPv3 session.
l2tp id	Configures the identifiers used by the local and remote provider edge routers at each end of an L2TPv3 session.

Feature Information for Multilink Frame Relay over L2TPv3/AToM

This feature enables Multilink Frame Relay switching over Layer 2 Tunnel Protocol Version 3 (L2TPv3) and Any Transport over MPLS (AToM). The feature works with like-to-like interfaces and disparate interfaces (L2VPN interworking).

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Multilink Frame Relay over L2TPv3/AToM

Feature Name	Releases	Feature Information
Multilink Frame Relay over L2TPv3/AToM	12.0(28)S 12.2(25)S 12.0(32)S 12.4(11)T	This feature was introduced in Cisco IOS Release 12.0(28)S for the Cisco 7200 and 7500 series routers. This feature was integrated into Cisco IOS Release 12.2(25)S. In Cisco IOS Release 12.0(32)S, this feature added support for the following pluggable modules for the Cisco 12000 series router: Cisco 4-port channelized T3 (DSO) shared port adapter, Cisco 8-port channelized T1/E1 shared port adapter, and the Cisco 1-port channelized OC-3/ STM-1 shared port adapter. This feature was integrated into Cisco IOS Release 12.4(11)T. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support.