Multilink Frame Relay over L2TPv3AToM

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

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

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, 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

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

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.
Step 3	**interface mfr** *number* **Example:**	Configures a multilink Frame Relay bundle interface and enters interface configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# interface mfr 1`	
Step 4	**frame-relay multilink bid** *name* **Example:** **Example:** `Router(config-if)# frame-relay multilink bid int1` **Example:**	(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**	Enables privileged EXEC mode.

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

	Command or Action	Purpose

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.

	Command or Action	Purpose
		• 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:** **Example:** **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:

Example:

```
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:

Example:

```
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

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

Standard	Title
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

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Multiprotocol Label Switching Command Reference* at http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List* .

- **xconnect**

Feature Information for Multilink Frame Relay over L2TPv3 AToM

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

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.