



Distributed Multilink Frame Relay FRF.16

Feature History

Release	Modification
12.0(24)S	The Distributed Multilink Frame Relay feature was introduced. This feature introduced Multilink Frame Relay (FRF.16) on VIP-enabled Cisco 7500 series routers.
12.3(4)T	This feature on VIP-enabled Cisco 7500 series routers was integrated into Cisco IOS Release 12.3(4)T.

This document describes the Distributed Multilink Frame Relay (dMFR) feature in Cisco IOS Release 12.0(24)S and Cisco IOS Release 12.3(4)T. The dMFR feature introduces MFR on VIP-enabled Cisco 7500 series routers. For information on MFR on other platforms, see the *Multilink Frame Relay (FRF.16)* document.

- [Finding Feature Information, page 2](#)
- [Feature Overview, page 2](#)
- [Supported Platforms, page 4](#)
- [Supported Standards MIBs and RFCs, page 5](#)
- [Prerequisites, page 6](#)
- [Configuration Tasks, page 6](#)
- [Monitoring and Maintaining Distributed Multilink Frame Relay, page 9](#)
- [Configuration Examples, page 9](#)
- [Glossary, page 9](#)

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.

Feature Overview

The Distributed Multilink Frame Relay feature introduces functionality based on the Frame Relay Forum Multilink Frame Relay UNI/NNI Implementation Agreement (FRF.16) to VIP-enabled Cisco 7500 series routers. The Distributed Multilink Frame Relay feature provides a cost-effective way to increase bandwidth for particular applications by enabling multiple serial links to be aggregated into a single bundle of bandwidth. Multilink Frame Relay is supported on User-to-Network Interfaces (UNI) and Network-to-Network Interfaces (NNI) in Frame Relay networks.

Multilink Frame Relay Bundles and Bundle Links

The Multilink Frame Relay feature enables you to create a virtual interface called a *bundle* or *bundle interface*. The bundle interface emulates a physical interface for the transport of frames. The Frame Relay data link runs on the bundle interface, and Frame Relay virtual circuits are built upon it.

The bundle is made up of multiple serial links, called *bundle links*. Each bundle link within a bundle corresponds to a physical interface. Bundle links are invisible to the Frame Relay data-link layer, so Frame Relay functionality cannot be configured on these interfaces. Regular Frame Relay functionality that you want to apply to these links must be configured on the bundle interface. Bundle links are visible to peer devices. The local router and peer devices exchange link integrity protocol control messages to determine which bundle links are operational and to synchronize which bundle links should be associated with which bundles.

Link Integrity Protocol Control Messages

For link management, each end of a bundle link follows the MFR Link Integrity Protocol and exchanges link control messages with its peer (the other end of the bundle link). To bring up a bundle link, both ends of the link must complete an exchange of ADD_LINK and ADD_LINK_ACK messages. To maintain the link, both ends periodically exchange HELLO and HELLO_ACK messages. This exchange of hello messages and acknowledgments serve as a keepalive mechanism for the link. If a router is sending hello messages but not receiving acknowledgments, it will resend the hello message up to a configured maximum number of times. If the router exhausts the maximum number of retries, the bundle link line protocol is considered down (unoperational).

The bundle link interface's line protocol status is considered up (operational) when the peer device acknowledges that it will use the same link for the bundle. The line protocol remains up when the peer device acknowledges the hello messages from the local router.

The bundle interface's line status becomes up when at least one bundle link has its line protocol status up. The bundle interface's line status goes down when the last bundle link is no longer in the up state. This behavior complies with the class A bandwidth requirement defined in FRF.16.

The bundle interface's line protocol status is considered up when the Frame Relay data-link layer at the local router and peer device synchronize using the Local Management Interface (LMI), when LMI is enabled. The bundle line protocol remains up as long as the LMI keepalives are successful.

Load Balancing

Distributed Multilink Frame Relay provides load balancing across the bundle links within a bundle. If a bundle link chosen for transmission happens to be busy transmitting a long packet, the load balancing mechanism can try another link, thus solving the problems seen when delay-sensitive packets have to wait.

Benefits

Flexible Pool of Bandwidth

By combining multiple physical interfaces into a bundle, you can design a Frame Relay interface with more bandwidth than is available from any single physical interface. For example, many new network applications require more bandwidth than is available on a T1 line. One option is to invest in a T3 line; however, T3 lines can be expensive and are not available in some locations. Distributed Multilink Frame Relay provides a cost-effective solution to this problem by allowing multiple T1 lines to be aggregated into a single bundle of bandwidth.

Greater Service Resilience When Links Fail

Greater service resilience is provided when multiple physical interfaces are provisioned as a single bundle. When a link fails, the bundle continues to support the Frame Relay service by transmitting across the remaining bundle links.

Restrictions

The Distributed Multilink Frame Relay feature has the following restrictions:

- ISDN interfaces and any type of virtual interfaces cannot be a bundle link.
- Distributed CEF is limited to IP traffic only; all other protocols are processed using the Route Switch Processor (RSP).
- Frame Relay fragmentation (FRF.12) is not supported.
- Multilink Frame Relay MIB (RFC 3020) is not supported.
- FRF.9 hardware compression over multilink Frame Relay is not supported.
- Each link in a bundle must reside on the same port adapter and all links in a bundle must have identical configurations. The same bandwidth for each link in the bundle is also recommended because bundles that contain individual links with different bandwidths process packets less efficiently.
- Fragmentation is not supported on the transmitting interface when used in conjunction with Distributed Multilink Frame Relay.

- The maximum differential delay is 50 ms.
- All T1 lines can be combined into one bundle. A VIP2-50 with 4 or 8 MB of SRAM supports up to 16 T1 bundles per VIP and a VIP2-50 with 2 MB of SRAM supports up to 8 T1 bundles per VIP. A maximum of 40 T1 bundles per VIP can be used on a VIP4-80.
- All E1 lines can be combined into one bundle. A VIP2-50 with 4 or 8 MB of SRAM supports up to 12 E1 bundles per VIP and a VIP2-50 with 2 MB of SRAM supports up to 8 E1 bundles per VIP. A maximum of 32 E1 bundles per VIP can be used on a VIP4-80.

Related Documents

- Cisco IOS Wide-Area Networking Configuration Guide , Release 12.3
- Cisco IOS Wide-Area Networking Command Reference , Release 12.3
- Multilink Frame Relay (FRF.16) (provides information on nondistributed Multilink Frame Relay)
- Cisco IOS Release 12.3 T command references (information on 12.0 S and 12.3 T commands)

Supported Platforms

- Cisco 7500 series router with a VIP2-50 or greater

This feature works on the following port adapters:

- PA-MC-T3
- PA-MC-2T3+
- PA-MC-E3
- PA-MC-2E1
- PA-MC-2T1
- PA-MC-4T1
- PA-MC-8T1
- PA-MC-8E1
- PA-MC-STM-1
- PA-MC-8TE1+
- PA-4T+
- PA-8T

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

<http://www.cisco.com/register> <http://www.cisco.com/register>

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards MIBs and RFCs

Standards

Multilink Frame Relay UNI/NNI Implementation Agreement (FRF.16.1), July 2001

MIBs

No new or modified MIBs are supported by this feature.

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

<http://tools.cisco.com/ITDIT/MIBS/servlet/index>

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

<http://www.cisco.com/register>

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

- Distributed Cisco Express Forwarding (CEF) must be enabled globally.
- Multilink Frame Relay must be configured on the peer device.
- The multilink Frame Relay peer device must not send frames that require assembly.

Configuration Tasks

Configuring a Multilink Frame Relay Bundle

To configure the bundle interface for Distributed Multilink Frame Relay, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **interface mfr** *number*
2. Router(config-if)# **frame-relay multilink bid** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# interface mfr <i>number</i> Example:	Configures a multilink Frame Relay bundle interface.
Step 2	Router(config-if)# frame-relay multilink bid <i>name</i> 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 shut and no shut commands in interface configuration mode.

Configuring a Multilink Frame Relay Bundle Link

To configure a bundle link interface for multilink Frame Relay, use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **interface serial** *number*
2. Router(config-if)# **encapsulation frame-relay mfr** *number[name]*
3. Router(config-if)# **frame-relay multilink lid** *name*
4. Router(config-if)# **frame-relay multilink hello** *seconds*
5. Router(config-if)# **frame-relay multilink ack** *seconds*
6. Router(config-if)# **frame-relay multilink retry** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# interface serial <i>number</i> Example:	Selects a physical interface and enters interface configuration mode.
Step 2	Router(config-if)# encapsulation frame-relay mfr <i>number[name]</i> Example:	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 3	Router(config-if)# frame-relay multilink lid <i>name</i> Example:	(Optional) Assigns a bundle link identification name to 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 shut and no shut commands in interface configuration mode.
Step 4	Router(config-if)# frame-relay multilink hello <i>seconds</i> Example:	(Optional) Configures the interval at which a bundle link will send out hello messages. The default value is 10 seconds.
Step 5	Router(config-if)# frame-relay multilink ack <i>seconds</i> Example:	(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 6	Router(config-if)# frame-relay multilink retry <i>number</i> Example:	(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.

Verifying Multilink Frame Relay

To verify multilink Frame Relay configuration, use the **show frame-relay multilink** command.

The following example shows output for the **show frame-relay multilink** command. Because a particular bundle or bundle link is not specified, information for all bundles and bundle links is displayed.

```
Router# show frame-relay multilink

Bundle: MFR0, state up, class A, no fragmentation
  ID: Bundle-Dallas
  Serial5/1, state up/up, ID: BL-Dallas-1
  Serial5/3, state up/add-sent, ID: BL-Dallas-3
Bundle: MFR1, state down, class B, fragmentation
  ID: Bundle-NewYork#1
  Serial3/0, state up/up, ID: BL-NewYork-1
  Serial3/2, state admin-down/idle, ID: BL-NewYork-2
```

The following example shows output for the **show frame-relay multilink** command with the **serial number** option. It displays information about the specified bundle link.

```
Router# show frame-relay multilink serial3/2
Bundle links :
Serial3/2, HW state :Administratively down, Protocol state :Down_idle, LID :Serial3/2
Bundle interface = MFR0,  BID = MFR0
```

The following examples show output for the **show frame-relay multilink** command with the **serial number** and **detail** options. Detailed information about the specified bundle links is displayed. The first example shows a bundle link in the "idle" state. The second example shows a bundle link in the "up" state.

```
Router# show frame-relay multilink serial3 detail
Bundle links:
Serial3, HW state = up, link state = Idle, LID = Serial3
Bundle interface = MFR0,  BID = MFR0
Cause code = none, Ack timer = 4, Hello timer = 10,
Max retry count = 2, Current count = 0,
Peer LID = Serial5/3, RTT = 0 ms
Statistics:
Add_link sent = 0, Add_link rcv'd = 10,
Add_link ack sent = 0, Add_link ack rcv'd = 0,
Add_link rej sent = 10, Add_link rej rcv'd = 0,
Remove_link sent = 0, Remove_link rcv'd = 0,
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
Hello sent = 0, Hello rcv'd = 0,
Hello ack sent = 0, Hello ack rcv'd = 0,
outgoing pak dropped = 0, incoming pak dropped = 0
```

```
Router# show frame-relay multilink serial3 detail
Bundle links:
Serial3, HW state = up, link state = Up, LID = Serial3
Bundle interface = MFR0,  BID = MFR0
Cause code = none, Ack timer = 4, Hello timer = 10,
Max retry count = 2, Current count = 0,
Peer LID = Serial5/3, RTT = 4 ms
Statistics:
Add_link sent = 1, Add_link rcv'd = 20,
Add_link ack sent = 1, Add_link ack rcv'd = 1,
Add_link rej sent = 19, Add_link rej rcv'd = 0,
Remove_link sent = 0, Remove_link rcv'd = 0,
Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
Hello sent = 0, Hello rcv'd = 1,
Hello ack sent = 1, Hello ack rcv'd = 0,
outgoing pak dropped = 0, incoming pak dropped = 0
```

Monitoring and Maintaining Distributed Multilink Frame Relay

To monitor and maintain Distributed Multilink Frame Relay, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
Router# debug frame-relay multilink [control [mfr number serial number]]	Displays debug messages for multilink Frame Relay bundles and bundle links.
Router# show frame-relay multilink [mfr number serial number] [detailed]	Displays configuration information and statistics about multilink Frame Relay bundles and bundle links.
Router# show interfaces mfr number	Displays information and packet statistics for the bundle interface.

Configuration Examples

Distributed Multilink Frame Relay Configuration Example

The following example shows the configuration of bundle "MFR1". Serial interfaces 5/0 and 6/0 are configured as bundle links.

```
interface MFR1
  frame-relay multilink bid first-bundle
  frame-relay traffic-shaping
  frame-relay class ocean
interface MFR1.1 point-to-point
  ip address 1.1.1.1 255.255.255.0
  frame-relay interface-dlci 100
interface Serial5/0
  encapsulation frame-relay MFR1
  frame-relay multilink lid first-link
  frame-relay multilink hello 9
  frame-relay multilink retry 3
interface Serial6/0
  encapsulation frame-relay MFR1
  frame-relay multilink ack 4
```

Glossary

BID --bundle identification. BID is the name used to identify the bundle. The BID can be assigned or the default can be used.

bundle --A logical grouping of one or more physical interfaces using the formats and procedures of multilink Frame Relay. A bundle emulates a physical interface to the Frame Relay data-link layer. The bundle is also referred to as the *mfr interface*.

bundle link --An individual physical interface that is a member of a bundle.

DLCI --data-link connection identifier. Value that identifies a permanent virtual circuit (PVC) in Frame Relay network.

HELLO message --Message that notifies a peer endpoint that the local endpoint is in the operational state (up).

HELLO_ACK --Message that notifies a peer endpoint that a hello message has been received.

LID --link identification. LID is the name used to identify a bundle link. The LID can be assigned or the default can be used.

LMI --Local Management Interface. Set of enhancements to the basic Frame Relay specification. LMI includes support for a keepalive mechanism, which verifies that data is flowing; a multicast mechanism, which provides the network server with its local DLCI and the multicast DLCI; global addressing, which gives DLCIs global rather than local significance in Frame Relay networks; and a status mechanism, which provides an ongoing status report on the DLCIs known to the switch.

NNI --Network-to-Network Interface. The interface between two Frame Relay devices that are both located in a private network or both located in a public network.

UNI --User-to-Network Interface. The interface between a Frame Relay device in a public network and a Frame Relay device in a private network.