

Frame Relay End-to-End Keepalive

This feature module describes the Frame Relay End-to-End Keepalive feature. It includes information on the benefits of the new feature, supported platforms, related documents, and so forth.

This document includes the following sections:

- Feature Overview, page 1
- Supported Platforms, page 3
- Supported Standards, MIBs, and RFCs, page 3
- Prerequisites, page 3
- Configuration Tasks, page 3
- Monitoring and Maintaining the Frame Relay End-to-End Keepalive Feature, page 4
- Configuration Examples, page 5
- Command Reference, page 7
- Debug Commands, page 21

Feature Overview

Frame Relay devices connect with each other through virtual circuits. Each virtual circuit is uniquely identified by a Data Link Connection Identifier (DLCI). In environments in which permanent virtual circuits (PVCs) are used, information regarding added or deleted PVCs and information about availability or unavailability of PVCs is carried through a Local Management Interface (LMI) with the use of status bits.

The Frame Relay switch within the local PVC segment deduces the status of the remote PVC segment through a Network-to-Network interface (NNI) and reports the status to the local router. If local management support within the switch is not end-to-end, the keepalive feature is the only source of information about the remote router. Frame Relay end-to-end keepalives provide status to verify that end-to-end communications are working and that traffic is getting through.

Each PVC connecting any two end devices has two separate keepalive systems. Two are needed because the upstream path may not be the same as the downstream path. One system sends out requests and handles responses to those requests. This system is called the send side. The other system handles and replies to requests from the device at the other end of the PVC. This second system is called the receive side. The send side on one device communicates with the receive side on the other device, and vice versa.

The send side sends out a keepalive request and waits for a reply to its request. If a reply is received before the timer expires, a send side:Frame Relay end-to-end keepalives is recorded. If no reply is received before the timer expires, an error event is recorded. A number of the most recently recorded events are examined. If enough error events are accumulated, the keepalive status of the VC is changed from up to down, or if enough consecutive successful replies are received, the keepalive status of the VC will be changed from down to up. The number of events that will be examined is called the event window.

The receive side is similar to the send side. The receive side waits for requests and send outs replies to those requests. If a request is received before the timer expires, a success event is recorded. If a request is not received, an error event is recorded. If enough error events occur in the event window, the state will be changed from up to down. If enough consecutive success events occur, the state will be changed from down to up.

The Frame Relay End-to-End Keepalive feature can be configured in one of four modes: bidirectional, request, reply, or passive-reply.

- In bidirectional mode, both the send side and receive side are enabled, and the device sends out and waits for replies to keepalive requests, and also waits for and replies to keepalive requests.
- In request mode, only the send side is enabled, and the device sends out and waits for replies to its keepalive requests.
- In reply mode, only the receive side is enabled, and the device waits for and replies to keepalive requests.
- In passive-reply mode, the device only responds to keepalive requests, but does not set any timers or keep track of any events.

Because end-to-end keepalives allow traffic flow in both directions, they can be used to carry control and configuration information from end-to-end. Consistency of information between end hosts is critical in applications such as those relating to prioritized traffic and Voice Over Frame Relay. While SVCs can convey such information within end-to-end signaling messages, PVCs will benefit from a bidirectional communication mechanism.

Benefits

- Enables monitoring of PVC status for network monitoring or backup applications
- Bi-directional communication of PVC status
- Configurable on a per PVC basis with configurable timers

Restrictions

The Frame Relay End-to-End Keepalive feature works between peer Cisco communications devices. It is derived from the Frame Relay LMI protocol. The key difference is that rather than run over the signaling channel, as is the case with LMI, the End-to-End Keepalive feature runs over individual data channels.

Encapsulation of keepalive packets is proprietary; therefore, the feature is available only on Cisco devices running a software release that supports the Frame Relay End-to-End Keepalive feature.

Related Documents

For additional information about configuring Frame Relay, refer to the Cisco IOS Release 12.0 *Wide-Area Networking Configuration Guide*

Supported Platforms

- Cisco 1005
- Cisco 1600 series
- Cisco 2500 series
- Cisco 3600 series
- Cisco 4000 series (Cisco 4000, 4000-M, 4500, 4500-M, 4700, 4700-M)
- Cisco 7000 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco MC3810

Supported Standards, MIBs, and RFCs

MIBs

No new or modified MIBs are supported by this feature.

For descriptions of supported MIBs and how to use MIBs, see the Cisco MIB web site on CCO at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Before configuring or using the Frame Relay End-to-End Keepalive feature, you must complete the following task:

- Configure Frame Relay (as described in the “Configuring Frame Relay” chapter of the Cisco IOS Release 12.0 *Wide-Area Networking Configuration Guide*).

Configuration Tasks

The Frame Relay End-to-End Keepalive feature is configured as one of four modes: bidirectional, request, reply, or passive-reply. Each end of a VC must be configured for this feature to function properly. After the feature has been configured, individual parameters can be modified if necessary.

See the following sections for configuration tasks for the Frame Relay End-to-End Keepalive feature. Each task in the list indicates if the task is optional or required:

- Configuring Frame Relay End-to-End Keepalive Mode (Required)
- Modifying Frame Relay End-to-End Parameters (Optional)

Configuring Frame Relay End-to-End Keepalive Mode

You must configure both ends of a VC to send keepalives. If one end is configured as bidirectional, the other end must also be configured as bidirectional. If one end is configured as request, the other end must be configured as reply or passive-reply. If one end is configured as reply or passive-reply, the other end must be configured as request.

Step	Command	Purpose
1	Router(config)# map-class frame-relay <i>map-class-name</i>	Specifies a map class for the VC.
2	Router(config-map-class)# frame-relay end-to-end keepalive mode { bidirectiona request reply passive-reply }	Specify the Frame Relay end-to-end keepalive mode.

The four modes determine the type of keepalive traffic each device sends and responds to:

- In bidirectional mode, the device will send keepalive requests to the other end of the VC and will respond to keepalive requests from the other end of the VC.
- In request mode, the device will send keepalive requests to the other end of the VC.
- In reply mode, the device will respond to keepalive requests from the other end of the VC.
- In passive-reply mode, the device will respond to keepalive requests from the other end of the VC, but will not track errors or successes.

Modifying Frame Relay End-to-End Parameters

If necessary, you can modify the default parameter values used by the Frame Relay End-to-End Keepalive feature. Use any of the following map-class configuration commands:

Command	Purpose
Router(config-map-class)# frame-relay end-to-end keepalive error-threshold { send receive } <i>count</i>	Modifies the number of errors needed to change the keepalive state from up to down.
Router(config-map-class)# frame-relay end-to-end keepalive event-window { send receive } <i>count</i>	Modifies the number of recent events to check for errors.
Router(config-map-class)# frame-relay end-to-end keepalive success-events { send receive } <i>count</i>	Modifies the number of consecutive success events required to change the keepalive state from down to up.
Router(config-map-class)# frame-relay end-to-end keepalive timer { send receive } <i>interval</i>	Modifies the timer interval.

Monitoring and Maintaining the Frame Relay End-to-End Keepalive Feature

Use the following command to monitor the status of the Frame Relay End-to-End Keepalive feature:

Command	Purpose
Router# show frame-relay end-to-end keepalive <i>interface</i>	Shows the status of Frame Relay end-to-end keepalive.

Configuration Examples

This section provides the following Frame Relay end-to-end keepalive configuration examples:

- Bidirectional Mode with Default Values Configuration Example
- Request Mode with Default Values Configuration Example
- Request Mode with Modified Values Configuration Example

Bidirectional Mode with Default Values Configuration Example

In the following example, the devices at each end of a VC are configured so that a DLCI is assigned to a Frame Relay serial interface, a map class is associated with the interface, and Frame Relay end-to-end keepalive is configured in bidirectional mode using default values:

```
! router1
router1(config) interface serial 0/0.1 point-to-point
router1(config-if) ip address 10.1.1.1 255.255.255.0
router1(config-if) frame-relay interface-dlci 16
router1(config-if) frame-relay class vcgrp1
router1(config-if) exit
!
router1(config)# map-class frame-relay vcgrp1
router1(config-map-class)# frame-relay end-to-end keepalive mode bidirectional

! router2
router2(config) interface serial 1/1.1 point-to-point
router2(config-if) ip address 10.1.1.2 255.255.255.0
router2(config-if) frame-relay interface-dlci 16
router2(config-if) frame-relay class vceek
router1(config-if) exit
!
router2(config)# map-class frame-relay vceek
router2(config-map-class)# frame-relay end-to-end keepalive mode bidirectional
```

Request Mode with Default Values Configuration Example

In the following example, the devices at each end of a VC are configured so that a DLCI is assigned to a Frame Relay serial interface and a map class is associated with the interface. One device is configured in request mode while the other end of the VC is configured in reply mode.

```
! router1
router1(config) interface serial 0/0.1 point-to-point
router1(config-if) ip address 10.1.1.1 255.255.255.0
router1(config-if) frame-relay interface-dlci 16
router1(config-if) frame-relay class eek
router1(config-if) exit
!
router1(config)# map-class frame-relay eek
router1(config-map-class)# frame-relay end-to-end keepalive mode request

! router2
router2(config) interface serial 1/1.1 point-to-point
router2(config-if) ip address 10.1.1.2 255.255.255.0
router2(config-if) frame-relay interface-dlci 16
router2(config-if) frame-relay class group_3
router1(config-if) exit
!
router2(config)# map-class frame-relay group_3
router2(config-map-class)# frame-relay end-to-end keepalive mode reply
```

Request Mode with Modified Values Configuration Example

In the following example, the devices at each end of a VC are configured so that a DLCI is assigned to a Frame Relay serial interface and a map class is associated with the interface. One device is configured in request mode while the other end of the VC is configured in reply mode. The event window, error threshold, and success events values are changed so that the interface will change state less frequently:

```
! router1
router1(config) interface serial 0/0.1 point-to-point
router1(config-if) ip address 10.1.1.1 255.255.255.0
router1(config-if) frame-relay interface-dlci 16
router1(config-if) frame-relay class eek
router1(config-if) exit
!
router1(config)# map-class frame-relay eek
router1(config-map-class)# frame-relay end-to-end keepalive mode request
router1(config-map-class)# frame-relay end-to-end keepalive event-window send 5
router1(config-map-class)# frame-relay end-to-end keepalive error-threshold send 3
router1(config-map-class)# frame-relay end-to-end keepalive success-events send 3

! router2
router2(config) interface serial 1/1.1 point-to-point
router2(config-if) ip address 10.1.1.2 255.255.255.0
router2(config-if) frame-relay interface-dlci 16
router2(config-if) frame-relay class group_3
router1(config-if) exit
!
router2(config)# map-class frame-relay group_3
router2(config-map-class)# frame-relay end-to-end keepalive mode reply
```

Command Reference

This section documents new commands. All other commands used with this feature are documented in the Cisco IOS Release 12.0 command reference publications.

- **frame-relay end-to-end keepalive error-threshold**
- **frame-relay end-to-end keepalive event-window**
- **frame-relay end-to-end keepalive mode**
- **frame-relay end-to-end keepalive success-events**
- **frame-relay end-to-end keepalive timer**
- **show frame-relay end-to-end keepalive**

In Cisco IOS Release 12.0(1)T or later, you can search and filter the output for **show** and **more** commands. This functionality is useful when you need to sort through large amounts of output, or if you want to exclude output that you do not need to see.

To use this functionality, enter a **show** or **more** command followed by the “pipe” character (|), one of the keywords **begin**, **include**, or **exclude**, and an expression that you want to search or filter on:

```
command | {begin | include | exclude} regular-expression
```

Following is an example of the **show atm vc** command in which you want the command output to begin with the first line where the expression “PeakRate” appears:

```
show atm vc | begin PeakRate
```

For more information on the search and filter functionality, refer to the Cisco IOS Release 12.0(1)T feature module titled *CLI String Search*.

frame-relay end-to-end keepalive error-threshold

To modify the keepalive error threshold value, use the **frame-relay end-to-end keepalive error-threshold** map-class configuration command. Use the **no** form of this command to reset the error threshold value to its default.

frame-relay end-to-end keepalive error-threshold {send | receive} *count*

no frame-relay end-to-end keepalive error-threshold {send | receive}

Syntax Description

send	The number of send-side errors within the event window that will cause the keepalive status to go from up to down.
receive	The number of receive-side errors within the event window that will cause the keepalive status to go from up to down.
<i>count</i>	Number of errors required. The maximum value is 32.

Defaults

The default value for both the send and receive error threshold is 2.

Command Modes

Map-class configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The send-side value can only be configured in bidirectional and request modes. The receive-side value can only be configured in bidirectional and reply modes. See the **frame-relay end-to-end keepalive mode** command

When you configure the error threshold, you will also want to configure the event window. See the **frame-relay end-to-end keepalive event-window** command.

Examples

The following example shows increasing the receive-side error threshold to 4 and changing the event window to 7:

```
map-class frame-relay olga
  frame-relay end-to-end keepalive reply
  frame-relay end-to-end keepalive error-threshold receive 4
  frame-relay end-to-end keepalive event-window receive 7
```

Related Commands

Command	Description
frame-relay end-to-end keepalive event-window	Configures the size of the event window.
frame-relay end-to-end keepalive mode	Configures the Frame Relay end-to-end keepalive mode.
frame-relay end-to-end keepalive success-events	Configures how many consecutive Frame Relay end-to-end keepalive success events must occur before the interface status is changed from down to up.
frame-relay end-to-end keepalive timer	Configures Frame Relay end-to-end keepalive timers.
map-class frame-relay	Defines a map class.
show frame-relay end-to-end keepalive	Shows status of Frame Relay end-to-end keepalive.

frame-relay end-to-end keepalive event-window

To modify the keepalive event window value, use the **frame-relay end-to-end keepalive event-window** map-class configuration command. Use the **no** form of this command to reset the event window value to its default.

frame-relay end-to-end keepalive event-window {send | receive} *size*

no frame-relay end-to-end keepalive event-window {send | receive}

Syntax Description

send	The size of the send-side event window.
receive	The size of the receive-side event window.
<i>size</i>	Number of events in the event window. The maximum value is 32.

Defaults

The default value for both the send and receive event windows is 3.

Command Modes

Map-class configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The send-side value can only be configured in bidirectional and request modes. The receive-side value can only be configured in bidirectional and reply modes. See the **frame-relay end-to-end keepalive mode** command

When you configure the event window, you will also want to configure the error-threshold. See the **frame-relay end-to-end keepalive error-threshold** command.

Examples

The following example shows increasing the receive-side error threshold to 4 and changing the event window to 7:

```
map-class frame-relay olga
  frame-relay end-to-end keepalive reply
  frame-relay end-to-end keepalive error-threshold receive 4
  frame-relay end-to-end keepalive event-window receive 7
```

Related Commands

Command	Description
frame-relay end-to-end keepalive error-threshold	Configures how many Frame Relay end-to-end keepalive errors must occur in the event window before the interface status is changed from up to down.
frame-relay end-to-end keepalive mode	Configures the Frame Relay end-to-end keepalive mode.
frame-relay end-to-end keepalive success-events	Configures how many consecutive Frame Relay end-to-end keepalive success events must occur before the interface status is changed from down to up.
frame-relay end-to-end keepalive timer	Configures Frame Relay end-to-end keepalive timers.
map-class frame-relay	Defines a map class.
show frame-relay end-to-end keepalive	Shows status of Frame Relay end-to-end keepalive.

frame-relay end-to-end keepalive mode

To enable Frame Relay end-to-end keepalives, use the **frame-relay end-to-end keepalive mode** map-class configuration command. Use the **no** form of this command to disable Frame Relay end-to-end keepalives.

```

frame-relay end-to-end keepalive mode {bidirectional | request | reply | passive-reply}
no frame-relay end-to-end keepalive
    
```

Syntax Description

bidirectional	Enables bidirectional mode.
request	Enables request mode.
reply	Enables reply mode.
passive-reply	Enables passive reply mode.

Defaults

When a Frame Relay end-to-end keepalive mode is enabled, default values depend on which mode is selected. For the meaning of the parameters, see the **frame-relay end-to-end keepalive timer**, **frame-relay end-to-end keepalive event-window**, **frame-relay end-to-end keepalive error-threshold**, and **frame-relay end-to-end keepalive success-events** commands.

Bidirectional Mode

Parameter	Send-Side	Receive-Side
Timer	10 seconds	15 seconds
Event Window	3	3
Error Threshold	2	2
Success Events	2	2

Request Mode

Parameter	Send-Side	Receive-Side
Timer	10 seconds	no value set
Event Window	3	no value set
Error Threshold	2	no value set
Success Events	2	no value set

Reply Mode

Parameter	Send-Side	Receive-Side
Timer	no value set	15 seconds
Event Window	no value set	3
Error Threshold	no value set	2
Success Events	no value set	2

Passive-Reply Mode

In passive-reply mode, no values are set.

Command Modes

Map-class configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

To enable Frame Relay end-to-end keepalives, Frame Relay must be configured. In addition, a map-class must be associated and a DLCI assigned to an interface, subinterface, VC or PVC. For more information on associating a frame-relay class with an interface, subinterface, VC or PVC, see the **frame-relay class** command. For more information on assigning a DLCI to an interface, subinterface, VC or PVC, see the **frame-relay interface-dlci** command.

In bidirectional mode, both ends of a VC send keepalive requests and respond to keepalive requests. If one end of the VC is configured in the bidirectional mode, the other end must also be configured in the bidirectional mode.

In request mode, the router sends keepalive requests and expects replies from the other end of the VC. If one end of a VC is configured in the request mode, the other end must be configured in the reply or passive-reply mode.

In reply mode, the router does not send keepalive requests, but waits for keepalive requests from the other end of the VC and replies to them. If no keepalive request has arrived within the timer interval, the router times out and increments the error counter by 1. If one end of a VC is configured in the reply mode, the other end must be configured in the request mode.

In passive-reply mode, the router does not send keepalive requests, but waits for keepalive requests from the other end of the VC and replies to them. No timer is set when in this mode, and the error counter is not incremented. If one end of a VC is configured in the passive-reply mode, the other end must be configured in the request mode.

Examples

The following example configures one end of a VC so that a DLCI is assigned to a Frame Relay serial interface, a map class is associated with the interface, and Frame Relay end-to-end keepalive is configured in bidirectional mode using default values:

```
router1(config) interface serial 0/0.1 point-to-point
router1(config-if) ip address 10.1.1.1 255.255.255.0
router1(config-if) frame-relay interface-dlci 16
router1(config-if) frame-relay class vcgrp1
router1(config-if) exit
!
router1(config)# map-class frame-relay vcgrp1
router1(config-map-class)# frame-relay end-to-end keepalive mode bidirectional
```

frame-relay end-to-end keepalive mode

The following example configures one end of a VC to reply to keepalive requests and to increment its error counter if no keepalive requests are received 30 seconds after the latest request:

```
map-class frame-relay oro34
  frame-relay end-to-end keepalive reply
  frame-relay end-to-end keepalive timer receive 30
```

Related Commands

Command	Description
frame-relay class	Associates a map class with an interface or subinterface.
frame-relay interface-dlci	Assigns a DLCI to a specified Frame Relay subinterface on the router or access server, or defines a specific PVC to a DLCI and applies a virtual template configuration for a PPP session.
frame-relay end-to-end keepalive error-threshold	Configures how many Frame Relay end-to-end keepalive errors must occur in the event window before the interface status is changed from up to down.
frame-relay end-to-end keepalive event-window	Configures the size of the event window.
frame-relay end-to-end keepalive success-events	Configures how many consecutive Frame Relay end-to-end keepalive success events must occur before the interface status is changed from down to up.
frame-relay end-to-end keepalive timer	Configures Frame Relay end-to-end keepalive timers.
map-class frame-relay	Defines a map class.
show frame-relay end-to-end keepalive	Shows status of Frame Relay end-to-end keepalive.

frame-relay end-to-end keepalive success-events

To modify the keepalive success events value, use the **frame-relay end-to-end keepalive success-events** map-class configuration command. Use the **no** form of this command to reset the success events value to its default.

```
frame-relay end-to-end keepalive success-events {send | receive} count
```

```
no frame-relay end-to-end keepalive success-events {send | receive}
```

Syntax Description

send	The number of consecutive send-side success events required to change the keepalive state from down to up.
receive	The number of consecutive receive-side success events required to change the keepalive state from down to up.
<i>count</i>	Number of consecutive success events required. The maximum value is 32.

Defaults

The default value for both the send and receive success events is 2.

Command Modes

Map-class configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The send-side value can only be configured in bidirectional and request modes. The receive-side value can only be configured in the bidirectional and reply modes. See the **frame-relay end-to-end keepalive mode** command.

If the success events value is set to a low value at the same time that a low value is set for the error threshold value of the **frame-relay end-to-end keepalive error-threshold** command, the keepalive state of the VC may flap from state to state.

Examples

The following example shows how to increase the success events value:

```
map-class frame-relay vcgrp4
  frame-relay end-to-end keepalive request
  frame-relay end-to-end keepalive success-events send 4
```

Related Commands

Command	Description
frame-relay end-to-end keepalive error-threshold	Configures how many Frame Relay end-to-end keepalive errors must occur in the event window before the interface status is changed from up to down.
frame-relay end-to-end keepalive event-window	Configures the size of the event window.
frame-relay end-to-end keepalive mode	Configures the Frame Relay end-to-end keepalive mode.
frame-relay end-to-end keepalive timer	Configures Frame Relay end-to-end keepalive timers.
map-class frame-relay	Defines a map class.
show frame-relay end-to-end keepalive	Shows status of Frame Relay end-to-end keepalive.

frame-relay end-to-end keepalive timer

To modify the keepalive timer value, use the **frame-relay end-to-end keepalive timer** map-class configuration command. Use the **no** form of this command to reset the timer value to its default.

```
frame-relay end-to-end keepalive timer {send | receive} interval
```

```
no frame-relay end-to-end keepalive timer {send | receive}
```

Syntax Description

send	How frequently to send a keepalive request.
receive	How long before the receive-side error counter is incremented if no request is received.
<i>interval</i>	Time in seconds for the timer to expire.

Defaults

The default value for the send timer is 10 seconds. The default value for the receive timer is 15 seconds.

Command Modes

Map-class configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The send-side value can only be configured in bidirectional and request modes. The receive-side value can only be configured in the bidirectional and reply modes. See the **frame-relay end-to-end keepalive mode** command

The send-side timer expires if a reply has not been received *interval* seconds after a request is sent. The receive-side timer expires if a request has not been received *interval* seconds after the previous request.

Examples

The following example shows how to set up one end of a VC to send a keepalive request every 15 seconds and increment the error counter if more than 22 seconds elapse between receiving keepalive responses:

```
map-class frame-relay vcgrp1
  frame-relay end-to-end keepalive bidirectional
  frame-relay end-to-end keepalive timer send 15
  frame-relay end-to-end keepalive timer receive 22
```

Related Commands

Command	Description
frame-relay end-to-end keepalive error-threshold	Configures how many Frame Relay end-to-end keepalive errors must occur in the event window before the interface status is changed from up to down.
frame-relay end-to-end keepalive event-window	Configures the size of the event window.
frame-relay end-to-end keepalive mode	Configures the Frame Relay end-to-end keepalive mode.
frame-relay end-to-end keepalive success-events	Configures how many consecutive Frame Relay end-to-end keepalive success events must occur before the interface status is changed from down to up.
map-class frame-relay	Defines a map class.
show frame-relay end-to-end keepalive	Shows status of Frame Relay end-to-end keepalive.

show frame-relay end-to-end keepalive

To display statistics about Frame Relay end-to-end keepalive, use the **show frame-relay end-to-end keepalive** EXEC command.

```
show frame-relay end-to-end keepalive [interface [DLCI]]
```

Syntax Description

interface (Optional) Interface to display.

DLCI (Optional) DLCI to display.

Defaults

If no interface specified, show all interfaces.

Command Modes

EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

Use this command to display the keepalive status of an interface.

Examples

The following example shows output from the **show frame-relay end-to-end keepalive** command:

```
Router# show frame-relay end-to-end keepalive interface s1

End-to-end Keepalive Statistics for Interface Serial11 (Frame Relay DTE)

DLCI = 100, DLCI USAGE = LOCAL, VC STATUS = STATIC (EEK UP)

SEND SIDE STATISTICS

Send Sequence Number: 86,          Receive Sequence Number: 87
Configured Event Window: 3,       Configured Error Threshold: 2
Total Observed Events: 90,        Total Observed Errors: 34
Monitored Events: 3,              Monitored Errors: 0
Successive Successes: 3,          End-to-end VC Status: UP

RECEIVE SIDE STATISTICS

Send Sequence Number: 88,          Receive Sequence Number: 87
Configured Event Window: 3,       Configured Error Threshold: 2
Total Observed Events: 90,        Total Observed Errors: 33
Monitored Events: 3,              Monitored Errors: 0
Successive Successes: 3,          End-to-end VC Status: UP
```

Related Commands

Command	Description
frame-relay end-to-end keepalive error-threshold	Configures how many Frame Relay end-to-end keepalive errors must occur in the event window before the interface status is changed from up to down.
frame-relay end-to-end keepalive event-window	Configures the size of the event window.
frame-relay end-to-end keepalive mode	Configures the Frame Relay end-to-end keepalive mode.
frame-relay end-to-end keepalive success-events	Configures how many consecutive Frame Relay end-to-end keepalive success events must occur before the interface status is changed from down to up.
frame-relay end-to-end keepalive timer	Configures Frame Relay end-to-end keepalive timers.

Debug Commands

This section documents the new **debug** command related to the Frame Relay End-to-End Keepalive feature.

- **debug frame-relay end-to-end keepalive**

debug frame-relay end-to-end keepalive

To display debug messages for the Frame Relay End-to-End Keepalive feature, use the **debug frame-relay end-to-end keepalive** command. Use the **no** form of this command to disable the display of debug messages.

[no] debug frame-relay end-to-end keepalive {events | packet}

Syntax Description

events Displays keepalive events.

packet Displays keepalive packets sent and received.

Defaults

This command is disabled by default.

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

We recommend that both commands be enabled.

Examples

The following examples show typical output from the **debug frame-relay end-to-end keepalive packet** command. The following example shows output for an outgoing request packet:

```
EEK (o, Serial0.1 DLCI 200): 1 1 1 3 2 4 3
```

The seven number fields that follow the colon signify the following:

Field	Description
first (example value = 1)	Information Element (IE) type.
second (example value = 1)	IE length.
third (example value = 1)	Report ID. 1 = request, 2 = reply.
fourth (example value = 3)	Next IE type. 3 = LIV ID (Keepalive ID).
fifth (example value = 2)	IE length. (This IE is a Keepalive IE.)
sixth (example value = 4)	Send sequence number.
seventh (example value = 3)	Receive sequence number.

The following example shows output for an incoming reply packet:

```
EEK (i, Serial0.1 DLCI 200): 1 1 2 3 2 4 4
```

The seven number fields that follow the colon signify the following:

Field	Description
first (example value = 1)	Information Element (IE) type.
second (example value = 1)	IE length.
third (example value = 2)	Report ID. 1 = request, 2 = reply.
fourth (example value = 3)	Next IE type. 3 = LIV ID (Keepalive ID).
fifth (example value = 2)	IE length. (This IE is a Keepalive IE.)
sixth (example value = 4)	Send sequence number.
seventh (example value = 4)	Receive sequence number.

The following example shows typical output from the **debug frame-relay end-to-end keepalive events** command:

```
EEK SUCCESS (request, Serial0.2 DLCI 400)
EEK SUCCESS (reply, Serial0.1 DLCI 200)
EEK sender timeout (Serial0.1 DLCI 200)
```

