IRIS® Frame Relay (IFR) Administrator’s Guide

Document Number 007-2913-001

CONTRIBUTORS

Written by Susan Thomas
Illustrated by Chris Everett
Document Production by Chris Everett
Engineering contributions by Robert Horen and Jay McCauley.

© 1995, 1996 Silicon Graphics, Inc.— All Rights Reserved
The contents of this document may not be copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

RESTRICTED RIGHTS LEGEND
Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and/or in similar or successor clauses in the FAR, or in the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94043-1389.

Silicon Graphics and IRIS are registered trademarks and IRIX is a trademark of Silicon Graphics, Inc.

IRIS® Frame Relay (IFR) Administrator’s Guide
Document Number 007-2913-001

Contents

List of Figures v

Introduction vii

Contents vii

Related Documentation and Resources vii

Typographical Conventions viii

Product Support viii

Suggested Reading viii

1. **What Is Frame Relay?** 1

How Frame Relay Works 2

Frame Relay Features 2

Prerequisites 3

Example Frame Relay Usage 3

Acronyms and Abbreviations 4

2. **Configuring and Starting Frame Relay** 7

Configuring the Frame Relay Network 8

- Configurable Parameters 8
  - IFR Subnetwork Parameters 9
  - IFR DLCI Parameters 11
  - MFE PVC Parameters 12
  - MFE Protocol Parameters 13

Starting and Stopping Frame Relay 14

Installation Verification 15

3. **Frame Relay Error Messages** 17

Index 19

List of Figures

Figure 1-1 Frame Relay in the Communications Hierarchy 1

Figure 1-2 Point-to-Point LAN Connection Using Frame Relay 3

Figure 1-3 Internet Connection Using Frame Relay 4

Introduction

This guide is intended for the administrator who is responsible for setting up and maintaining the Frame Relay software. It provides information about how to set up the Frame Relay software as well as descriptions of the configuration parameters and files that are used to run Frame Relay. This guide describes the steps needed to start Frame Relay and how to modify the configuration parameters, if necessary. It also describes how to verify the installation is working correctly and provides descriptions of associated error messages.

Contents

The *IRIS Frame Relay (IFR) Administrator's Guide* contains:

Chapter 1 “What Is Frame Relay?” introduces the fundamentals of Frame Relay operation, describes the key features, and provides an example of its real world usage.

Chapter 2 “Configuring and Starting Frame Relay” explains how to configure the parameters that determine how Frame Relay operates, how to start and stop Frame Relay, and how verify a correct installation.

Chapter 3 “Frame Relay Error Messages” provides the error messages associated with Frame Relay.

Related Documentation and Resources

The following provide additional information and resources for use with Frame Relay.

- The reference manual pages provide concise reference information for the use of commands, template configuration file formats, and driver descriptions. The reference pages are available online with the man(1) command.
- The Frame Relay release notes provide installation instructions and specific information about the current release. Exceptions to this guide are also provided.

- The operating system guides you received with your computer system contain information pertaining to system kernel reconfiguration.
- The installation guide for your synchronous communications board contain information about your board and its configuration.

Typographical Conventions

These type conventions and symbols are used in this guide:

Italics Filenames, IRIX™ commands, variables, IRIX command arguments, command flags, titles of publications, icon names

`Screen type` Code examples, file excerpts, and screen displays (including error messages)

`Bold Screen type`
User input

() (Parentheses) Following IRIX commands, they surround the reference page (man page) section where the command is described

# IRIX shell prompt for the superuser (*root*)

Product Support

Silicon Graphics offers a comprehensive product support and maintenance program for their products. For information about using support services for this product, refer to the release notes that accompany it.

Suggested Reading

ITU-T Recommendation Q.922 — ISDN Data Link Layer Specification for Frame Mode Bearer Services

ITU-T Recommendation Q.933 — DSS1 Signalling Specification for Frame Mode Basic Call Control

FRF.1 — Frame Relay Forum User to Network Implementation Agreement

FRF.3 — Frame Relay Forum Multiprotocol Encapsulation Implementation Agreement

Frame Relay Specification With Extensions, Revision 1.0 (LMI Specification)

ISO/IEC TR 9577:1992 — Protocol Identification in the Network Layer

RFC 1490 — Multiprotocol Interconnect over Frame Relay

RFC 1293 — Inverse Address Resolution Protocol

UNIX International — Data Link Provider Interface Specification, Revision 2.0.0 (August 1991)

Chapter 1

What Is Frame Relay?

This chapter provides an overview of IRIS Frame Relay (IFR), including a description of how it works, what it needs to run, and some examples of how it is commonly used.

Frame Relay is software that allows two LANs to be connected via a synchronous communication line. Frame Relay software manages the interface between the IP protocol and the lower-level synchronous communications board driver, as shown in Figure 1-1.

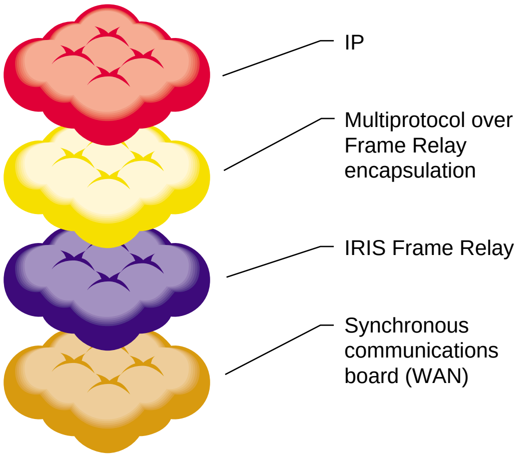
IP
Multiprotocol over Frame Relay encapsulation
IRIS Frame Relay
Synchronous communications board (WAN)

Figure 1-1 Frame Relay in the Communications Hierarchy

How Frame Relay Works

Frame Relay, residing at the data link layer, passes frames of data through a synchronous line to another site. The frames of data are managed by the configuration parameters that are chosen and contained in the configuration files.

Frame Relay is accessed by Link Access Procedures for Frame Relay (LAPF). LAPF conveys Data Link Service Data Units (DLSDUs) between DL-Service Users across the user-network interface over synchronous lines. Frame mode bearer connections are established using procedures specified in either Recommendation Q.933 or (for PVCs) by subscription.

Frame Relay Features

The key features of Frame Relay are:

- symmetric procedural behavior with respect to the User Network Interface (UNI), allowing direct user-to-user internetworking with the network side being passive (or only supporting the DL-CORE protocol)
- DL-CORE subset of LAPF
- use within a layered protocol suite, which allows internetworking between frame relaying, frame switching, and IP-based services
- MFE—Multiprotocol over Frame Relay Encapsulation according to RFC 1490, FRF.3, and T1.617 Annex F
- congestion control
- PVC status procedures (T1.617 Annex D/Q.933 Annex A/Original Group of Four)

Prerequisites

Frame Relay requires the following hardware and software:

- IRIX 6.2
- synchronous interface driver
- synchronous communications board

Example Frame Relay Usage

Frame Relay can be used to connect two LANs. The LANs may be connected by a Point-to-Point dedicated leased line, as shown in Figure 1-2, or through the Internet, as shown in Figure 1-3. The connection, using a synchronous communications board, supports several lines, each allowing data to be passed at speeds from 56 Kbps to a maximum of 2 Mbps. Frame Relay's high-speed data rate might be beneficial in connecting two Web servers.

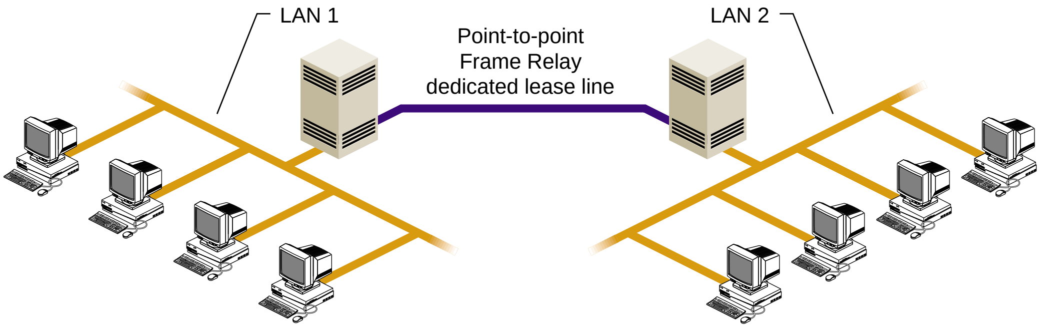

Figure 1-2 Point-to-Point LAN Connection Using Frame Relay

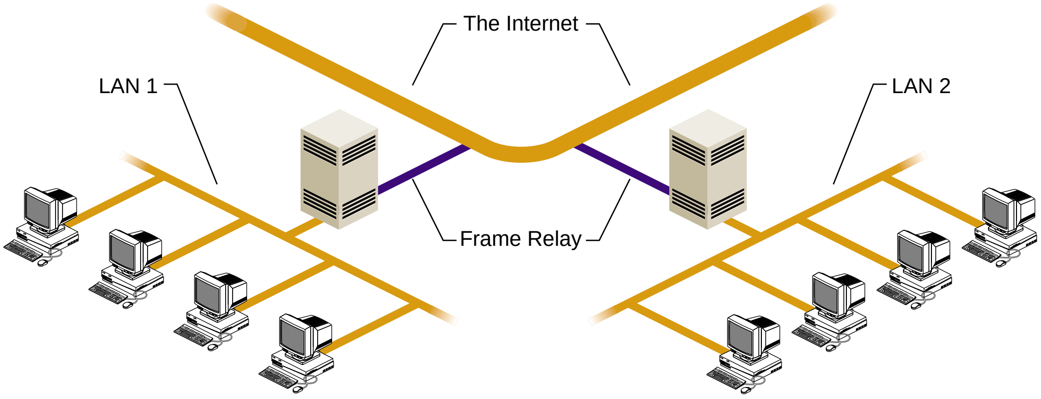

Figure 1-3 Internet Connection Using Frame Relay

Acronyms and Abbreviations

The following acronyms are used throughout this guide.

ANSI	American National Standards Institute
BECN	Backward Explicit Congestion Notification
CEI	Connection endpoint identifier
CLNP	Connectionless Network Protocol
DE	Discard Eligible
DLCI	Data Link Connection Identifier
DLPI	Data Link Provider Interface
DLSAP	Data Link Service Access Point
DLSDU	Data Link Service Data Unit
DL-CORE	Core aspects of LAPF
FECN	Forward Explicit Congestion Notification
FR	Frame Relay

FRF	Frame Relay Forum
IFR	The Frame Relay DL-CORE (Frame Relay Service) driver
IP	Internet Protocol
ITU-T	International Telecommunications Union - Telecommunications
LAPF	Link Access Procedures for Frame Mode Bearer Services
MC	Communication between layer management and DL-CORE
MDL	Communication between layer management and the data link layer
MFE	Multiprotocol over Frame Relay Encapsulation
NLPID	Network Layer Protocol Identifier
OUI	Organizationally Unique Identifier
PDU	Protocol Data Unit
PID	Protocol Identifier
PPA	Physical Point of Attachment
PVC	Permanent (or semi-permanent) Frame Relay connection
RFC	Request For Comments
SNAP	Subnetwork Access Protocol
SNID	Subnetwork Identifier
UI	Unnumbered Information
VC	Virtual Circuit
VCID	VC Identifier
WAN	WAN driver
XID	Exchange Identification

Chapter 2

Configuring and Starting Frame Relay

This chapter describes how to configure and start Frame Relay. It provides descriptions of the commands, configuration files, and configuration parameters within the configuration files. The sections are:

- "Configuring the Frame Relay Network"—the configurable parameters that can be used to dynamically tune the network protocol drivers. This tuning may take place at network startup time and, in some cases, while the network is already up.
- "Starting and Stopping Frame Relay"—how Frame Relay is controlled by the network daemon *snetd*.
- "Installation Verification"—how to verify that your installation and configuration is correct.

Additional information on Frame Relay is available in the online reference pages, which describe all programs, utilities, and system files. References to these pages include the section number, for example, ifrstat(1M).

Configuring the Frame Relay Network

The Frame Relay network configuration is held in the file */etc/config/snetd.options*. The following example shows the MFE driver is configured to support the IP network layer protocols.

```
wan01    d     /dev/vsc/wan01
wan03    d     /dev/vsc/wan03
wan11    d     /dev/vsc/wan11
wan13    d     /dev/vsc/wan13
wan21    d     /dev/vsc/wan21
wan23    d     /dev/vsc/wan23
wan31    d     /dev/vsc/wan31
wan33    d     /dev/vsc/wan33
ip_mfe   dc    /dev/ip_mfe
ifr      dc    /dev/ifr
mfe      dc    /dev/mfe
%%

ip_mfe   mfe
IP_NET={192.26.76.10,255.255.255.0,forwb=n,kalive=y,1500,fr,192.26.76.12} \
                  SHELL="mfevc -sB -c40 -d /dev/ip_mfe -A def.mfe"
mfe      ifr      SHELL="mfevc -sB -c40 -A def.mfe" \
                  SHELL="mfeproto -p40 -d /dev/mfe -A ip.mfe"
ifr      wan31    IFR_SET_SNID=B \
                  SHELL="ifrtune -sB -P def.dte.ifr" \
                  SHELL="ifrdlci -sB -c40 -A def.dlci"
```

In this example, *snetd* configures the network as follows:

- Configures MFE to support the IP network layer protocols.
- Links a single stream between IFR and the WAN driver and assigns it the subnetwork identifier B.
- Configures the stream with the default WAN and IFR subnetwork parameters.
- Links a single stream between MFE and IFR and binds them to VCID 40 on subnetwork B.

Configurable Parameters

Frame Relay implements a comprehensive range of configurable parameters for IFR and MFE. These are described in this section.

The parameters are held in template files. Most of the parameters are configurable and can be changed by editing the file with a text editor.

Note: It is recommended that you use the default parameters, as supplied by Silicon Graphics. Unless you are familiar with Frame Relay or are working under the direction of Silicon Graphics support personnel, care should be used when modifying parameters.

Using files to store parameters has two advantages:

- You can switch quickly between different network setups, if the files already exist, without having to directly modify any parameters.
- If you do have to change one parameter, there’s no need to set all the rest. You can keep their existing values.

A number of default template files are provided with Frame Relay. These are:

def.dce.ifr	IFR default configuration for the network side
def.dte.ifr	IFR default configuration for the user side
def.dlci	IFR default PVC configuration
def.mfe	MFE default PVC configuration
ip.mfe	MFE protocol configuration for IP

You can also create your own files, based on the template files, to suit your own network topologies.

All template files, default or otherwise, must be located in the directory */var/opt/snet/template*.

IFR Subnetwork Parameters

The IFR subnetwork parameters are described in ifrtplate(4). Each stream between IFR and the WAN driver represents a subnetwork. IFR subnetworks are configured using the ifrtune(1M) utility.

DTE/DCE IFR can be configured to act as either the user side (*DTE*) or the network side (*DCE*). The network side is used for testing since there is no switching functionality.

Permitted values are U for the user side and N for the network side. Typically, you should pick the U value (DTE).

DLCILENGTH The length (in bytes) of the Frame Relay address field (DLCI). In this release, 2 is the only value supported.

LMIDLCI The DLCI value used by the Local Management Interface (LMI). This value is used by IFR to recognize any LMI frames it receives.

A value of 0 means use the DLCI value appropriate for the standard in use. Other values are for testing purposes only.

T391 Link integrity verification polling timer. This is the time period (in seconds) between STATUS ENQUIRY messages sent by IFR acting as the user side. Status inquiry messages are sent by IFR to the network as part of the LMI operation. This time period is known as the "polling interval."

The default value is 10.

T392 Polling verification timer. This is a time limit for the network to respond to a STATUS ENQUIRY. It must be set greater than *T391*.

N391 Full status polling counter. Every *N391* polling cycles, IFR requests a FULL STATUS ENQUIRY from the network of all PVCs. The larger the value, the longer it may take before IFR becomes available to newly created or deleted PVCs.

The default value is 6.

N392 Error threshold value. If the number of errors detected by IFR reaches the threshold value within the last *N393* events, then IFR can assume that there is a service-affecting condition at the user-network interface. IFR stops transmitting data and continues link verification procedures. IFR detects service restoration by detecting that *N392* consecutive events have occurred without error.

The default value is 3.

N393 Monitored events count. This value is used as a measuring period for counting consecutive error or non-error events. It allows IFR to detect whether error or non-error events predominate over a given period.

The default value is 4.

MAXFRAMESIZE

The maximum size of a frame (in bytes) that can be accepted by IFR. Any frames larger than this maximum size are discarded. The value should be the same as the WAN maximum frame size.

ACCESSRATE The access rate of the physical interface. This is the maximum amount of data (bits per second) that the physical interface can transmit. This parameter is not used in IFR.

STANDARD Local management interface standard. The type of LMI to be used by IFR. The same type of LMI must also be used at the other end of the link.

Permitted values are:

0 ITU-T Q.933 Annex A

1 Current version of ANSI T1.617 Annex D

2 1991 version of ANSI T1.617 Annex D

3 Original Group of Four

Note: The only difference between the current and 1991 ANSI standards is the value of the link integrity verification element identifier. It is defined to be 0x03 in the current version and 0x19 in the 1991 version.

IFR DLCI Parameters

The IFR DLCI parameters are described in ifrdlciconf(4). IFR can support multiple PVCs (there is a compile-time configurable limit), each being identified by its DLCI. PVCs are configured using the ifrdlci(1M) utility.

CIR Committed information rate. The committed rate (in bits per second) at which the Frame Relay network accepts information from the user system under normal conditions. IFR uses this value to flow control PVC users. This value must be configured in accordance with your line speed.

Bc Committed burst size. The maximum amount of data (in bits) that the Frame Relay network agrees to transfer, under normal conditions, during a measurement interval. This data may or may not be contiguous (that is, it may appear in one or more frames).

The measurement interval is defined as *T* = *Bc*/*CIR*, where *CIR* is the committed information rate and *Bc* is the committed burst size. This value must be configured in accordance with your line speed. Typically, *Bc* is 1/10 of *CIR*.

Be Excess burst size. The maximum amount of data (in bits), in excess of the committed burst size, that the Frame Relay network attempts to deliver during a measurement interval. This data may or may not be contiguous (that is, it may appear in one or more frames). Excess burst is marked discard–eligible (DE bit) by the IFR driver. This value must be configured in accordance with your line speed.

If set to 0, IFR will not mark any data as discard-eligible.

STEPCOUNT Step count. The value is used by IFR when transmitting frames to increase or reduce the committed information rate. If IFR receives this number of consecutive frames with the BECN bit set, it reduces its CIR to the “next step” rate below the current offered rate. If it receives this number of consecutive frames and the BECN bit is not set, it increases its CIR.

FLOWSTYLE Flow control style. This value determines the type of congestion control used by IFR. The value is a bit mask, so both types can be enabled by adding the values (for example, 3 enables FECN and BECN).

FECN-based control alters the *CIR* based on the number of frames received with the FECN bit set versus the number of frames received without the FECN bit set over a given period.

BECN-based control alters the *CIR* based on the number of consecutive frames received with or without the BECN bit set during a given period.

Permitted values are:

0	No flow control
1	FECN
2	BECN

MFE PVC Parameters

The MFE PVC parameters are described in mfetemplate(4). Each stream between MFE and IFR represents a PVC to be used for multiprotocol encapsulation and is configured with the mfevc(1M) utility.

ADRLEN Address length. The length (in bytes) of the Frame Relay address field (DLCI) to assume if no other information is received from the IFR driver.

Only 2-byte addresses are supported by IFR.

MAXSDU Maximum DLSDU size. The maximum Data Link Service Data Unit (DLSDU) size (in bytes) to assume no other information is received from the IFR driver. Any maximum DLSDU size indicated by IFR is always used in preference. This value is used by MFE to determine when to fragment outgoing PDUs.

MAXPDU Maximum PDU size. The maximum size (in bytes) of a PDU that can be reassembled by MFE. This value is used to limit the length of the reassembly queue for incoming fragmented PDUs. PDUs longer than this are discarded.

FRGSEQ Sequence number. The initial sequence number to use in outgoing fragmented PDUs. A value of 0 means choose a random value, which is the recommended behavior.

MFE Protocol Parameters

The MFE protocol parameters are described in mfeprotoconf(4). MFE can support a number of protocols (there is a compile-time upper limit). Protocols are configured using the mfeproto(1M) utility.

MINPDUSZ Minimum PDU size. The minimum PDU size required by the network layer. The actual value conveyed by MFE to the upper datalink service user is the greater of this value and the maximum DLSDU size supported by the attached subnetwork or virtual circuit, less the MFE encapsulation header length. This value is configurable to support network layer protocols, which require a reasonably large maximum DLSDU size. For example, the OSI IS-IS routing protocol requires at least 512 bytes to work, which means some IS-IS PDUs are fragmented and reassembled by MFE if the maximum DLSDU size supported by the network is less than 512 bytes.

ENCAPS Encapsulation header. The encapsulation header (as a hexadecimal string) to prepend to all outgoing PDUs. The string must have an even number of digits. A zero length string is indicated by two dashes (- -). OSI PDUs have no encapsulation header; IP packets use CC.

RETAIN An indication as to whether or not the encapsulation header should be retained in incoming PDUs. Use Y to retain the header, N to remove it. OSI PDUs retain their headers; IP packets do not.

COUNT The number of encapsulation headers to be used for demultiplexing. Up to **3** are supported. This allows OSI CLNP, ES-IS, and IS-IS PDUs to be received on the same upper stream.

DEMUX Demultiplexing header. The encapsulation headers (as hexadecimal strings) to be used for demultiplexing. The strings must have an even number of digits. A zero-length string is indicated by two dashes. Only the first *Count* strings are significant.

Starting and Stopping Frame Relay

The Frame Relay networking software is initialized by a network daemon program called *snetd* (see snetd(1M)). *snetd* links the network drivers and configures them, as defined by various Frame Relay network configuration files. It supports the network by keeping open the STREAMS file descriptor(s) used for linking the stack.

The *snetd* daemon sleeps silently while the network is up.

Killing *snetd* closes all STREAMS file descriptors used for linking the stack, reinitializes all Frame Relay kernel data structures, and frees any resources.

Note: Killing *snetd* also kills all other communications services it supports such as X.25, SNA, and Frame Relay.

To start Frame Relay, follow these steps:

1. Change directories to */etc/config*. Type:

   ```
   cd /etc/config
   ```

2. Edit the file *snetd.options* to reflect your site configuration. Refer to “Configuring the Frame Relay Network” on page 8 if necessary.
3. Start the *snetd* daemon. Type:

   ```
   snetd
   ```

   The *snetd* daemon reads the file *snetd.options* and uses the configuration parameters to start Frame Relay.

Installation Verification

This section describes how to verify a correct installation and check that IFR is correctly communicating with the remote site.

The installed software can be verified using the *netstat* command, as follows:

```
#netstat -ia
```

An example of the output, based on the sample configuration in the *snetd.options* file used in Chapter 2, follows:

```
Name Mtu   Network     Address           Ipkts Ierrs    Opkts Oerrs  Coll
et0  1500  150.166.43  whoopi.engr.sgi   20002     0     3849     0    32
                        allhosts-mcast
                        08:00:69:02:4b:b6
fv0  4472  150.166.13  tr-whoopi.engr.    1466     0     1449     1     0
                        allhosts-mcast
                        40:00:70:00:00:02
lo0  8304  loopback    localhost          2538     0     2538     0     0
                        allhosts-mcast
fr0  1500  (pt-to-pt)  fr-barrow.engr      426     0     1324     0     0
                        allhosts-mcast
```

The *ifconfig* command can be used to verify the setup of the Frame Relay interface, fr0, as follows.

```
#ifconfig fr0
```

An example of the output, based on the sample configuration *snetd.options* file used in Chapter 2, follows:

```
fr0: flags=851<UP,POINTOPOINT,RUNNING,MULTICAST>
        inet 192.26.76.10 --> 192.26.76.12 netmask 0xffffff00
```

The ifrstat(1M) utility can be used to check the status with the remote site. The following output shows the network is OK and two PVCs are ACTIVE. The values in the txpkts and rxpkts columns should be incrementing if Frame Relay is running correctly.

```
#../bin/ifrstat -sA -S
```

```
#################################################################################
########
#                               Statistics for PPA A #
#################################################################################
########
PPA        txpkts     rxpkts     txbytes    rxbytes    wanflows   date cleared
A          10         10         140        160        0          May 15 15:41:05
rxtoobig   rxinvDLCI  rxunaDLCI  txstops    txnobuffs  txinvrq    rxdrops    rxin
vrq
0          0          0          0          0          0          0          0
LMItxpoll  LMIwnflow  rxfull     rxseqonly  rxasynch   rxBECN     rxLMIerr   LMIt
imout
10         0          2          8          0          0          0          0
LMIrxpoll  txfull     txseqonly  txasynch   txBECN     rxpollerr
0          0          0          0          0          0
#################################################################################
########
#                               Status Report for PPA A #
#################################################################################
########
Network status (LMI):ALL OK
DLCIs :
       40 : ACTIVE
       32 : ACTIVE
```

You can check on the current configuration of a particular network driver by running the appropriate tuner application (see ifrtune(1M), ifrdlci(1M), mfevc(1M), mfeproto(1M)).

Chapter 3

Frame Relay Error Messages

This chapter provides a listing of error messages. Should you see any of these errors on your system console or in */usr/adm SYSLOG*, please report the problem to your Silicon Graphics support person immediately. No corrective active should be attempted without assistance from your Silicon Graphics support person.

```
ip_mfe_lower_proto:  bad message format
ip_mfe_lower_proto:  dl_error_ack <error code>
ip_mfe_lower_proto:  dl_uderror_ack <error code>
ip_mfe_lower_proto:  unknown primitive <command code>
ip_mfe_ok_ack:  bad message format
ip_mfe_ok_ack:  bind allocb failed
ip_mfe_ok_ack:  no buffers
ip_mfe_ok_ack:  detach allocb failed
ip_mfe_ok_ack:  invalid state <bad state number>
ip_mfe_unitdata_ind:  data missing
ip_mfelrput:  db_type = <bad type>
ip_mfe_do_ifoutput:  allocb failed
ip_mfe:  bad message format for addvc
ip_mfe:  format_attach failed
ip_mfe:  format_unbind failed
ip_mfe:  M_IOCNAK  ioc_cmd = <command code>  ioc_error = <error code>
```

Index

A

ACCESSRATE IFR subnetwork parameter, 11
acronyms, 4
additional resources, vii
address length, 12
ADRLEN MFE PVC parameters, 12

B

Bc IFR DLCI parameter, 11
Be IFR DLCI parameter, 11

C

CIR IFR DLCI parameter, 11
committed burst size (Bc), 11
committed information rate (CIR), 11
configurable parameters, 8-14
configuration template files, 9
configuring Frame Relay network, 8
connecting LANs, 1, 3
conventions, typographical, viii
COUNT MFE protocol parameter, 13

D

data link layer, 2
default template files, 9
def.dce.ifr template file, 9
def.dlci template file, 9
def.dte.ifr template file, 9
def.mfe template file, 9
demultiplexing header., 14
DEMUX MFE protocol parameter, 14
DLCILENGTH IFR subnetwork parameter, 10
DTE/DCE IFR subnetwork parameter, 9

E

ENCAPS MFE protocol parameter, 13
encapsulation header, 13
error messages, 17
/etc/config/snetd.options, 8
Excess burst size (Be), 11

F

FLOWSTYLE IFR DLCI parameter, 12
frame mode bearer connections, 2
Frame Relay features, 2
FRGSEQ MFE PVC parameters, 13

H

hardware prerequisites, 3

I

ifconfig command, 15
ifrdlci(1M), 11, 16
ifrdlciconf(4), 11
IFR DLCI parameters, 11
IFR subnetwork parameters, 9
ifrtplate(4), 9
ifrtune(1M), 9, 16
installation verification, 14
ip.mfe template file, 9

L

LANs, connecting, 1, 3
link integrity verification polling timer, 10
LMIDLCI IFR subnetwork parameter, 10
Local Management Interface (LMI), 10
local management interface standard, 11

M

MAXFRAMESIZE IFR subnetwork parameter, 10
maximum DLSDU size, 13
maximum frame size, 10
maximum PDU size, 13
MAXPDU MFE PVC parameters, 13
MAXSDU MFE PVC parameters, 13
mfeproto(1M), 13, 16
MFE protocol parameters, 13
mfeprotoconf(4), 13
MFE PVC parameters, 12
mfevc(1M), 16
minimum PDU size, 13
MINPDUSZ MFE protocol parameter, 13

N

N391 IFR subnetwork parameter, 10
N392 IFR subnetwork parameter, 10
N393 IFR subnetwork parameter, 10
netstat, 15
netstat command, 15
network cloud connection, 4
network configuration, 8
network configuration file, *snetd.options*, 8

O

online reference pages, 7
overview of Frame Relay, 1

P

point-to-point connections, 3
point-to-point LAN connection, 3
polling interval, 10
prerequisites, 3
product support, viii

R

related documentation, vii
RETAIN MFE protocol parameter, 13

S

sequence number, 13
snetd(1M), 14
snetd daemon, 14
snetd.options configuration file, 8, 14
software prerequisites, 3
STANDARD IFR subnetwork parameter, 11
starting Frame Relay, 14
STATUS ENQUIRY messages, 10
STEPCOUNT IFR DLCI parameter, 12
stopping Frame Relay, 14
STREAMS file descriptors, 14
synchronous communication line, connecting, 1
synchronous communications board, 3
synchronous communications board driver, 1

T

T391 IFR subnetwork parameter, 10
T392 IFR subnetwork parameter, 10
template files, 9
typographical conventions, viii

U

/usr/adm SYSLOG, 17

V

verifying installtion, 14

Tell Us About This Manual

As a user of Silicon Graphics documentation, your comments are important to us. They help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics to comment on:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Important Note

Please include the title and part number of the document you are commenting on. The part number for this document is
007-2913-001.

Thank you!

Three Ways to Reach Us

The **postcard** opposite this page has space for your comments. Write your comments on the postage-paid card for your country, then detach and mail it. If your country is not listed, either use the international card and apply the necessary postage or use electronic mail or FAX for your reply.

If **electronic mail** is available to you, write your comments in an e-mail message and mail it to either of these addresses:

- If you are on the Internet, use this address: techpubs@sgi.com
- For UUCP mail, use this address through any backbone site:
  [your_site]!sgi!techpubs

You can forward your comments (or annotated copies of pages from the manual) to Technical Publications at this **FAX** number:

415 965-0964