Nways Multiprotocol Access Services

Software User's Guide
Version 1 Release 1

IBM

Nways Multiprotocol Access Services

Software User's Guide
Version 1 Release 1

> **Note**
>
> Before using this document, read the general information under "Notices" on page xxi.

First Edition (March 1997)

This edition applies to Version 1 Release 1 of the IBM Nways Multiprotocol Access Services and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments is provided at the back of this publication. If the form has been removed, you may address your comments to:

    Department CGF
    Design & Information Development
    IBM Corporation
    P.O. Box 12195
    RESEARCH TRIANGLE PARK NC 27709
    USA

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking	IBM	RISC System/6000
AIX	Micro Channel	System/370
AIXwindows	NetView	VTAM
APPN	Nways	
BookManager	PS/2	

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Preface

This manual contains the information that you will need to use the router user interface for configuration and operation of the IBM Nways Multiprotocol Access Services base code installed on your Nways device. With the help of this manual, you should be able to perform the following processes and operations:

- Configure, monitor, and use the IBM Nways Multiprotocol Access Services base code.

- Configure, monitor, and use the interfaces and Link Layer software supported by your Nways device.

This manual contains the information you will need to configure bridging and routing functions on an Nways device. The manual describes all of the features and functions that are in the software. A specific Nways device might not support all of the features and functions described. If a feature or function is device-specific, a notice in the relevant chapter or section indicates that restriction.

This manual supports the IBM 2216 and refers to this product as either "the routers" or "the device". The examples in the manual represent the configuration of an IBM 2216 but the actual output you see may vary. Use the examples as a guideline to what you might see while configuring your device.

Who Should Read This Manual: This manual is intended for persons who install and manage computer networks. Although experience with computer networking hardware and software is helpful, you do not need programming experience to use the protocol software.

To Get Additional Information: Changes may be made to the documentation after the books are printed. If additional information is available or if changes are required after the books have been printed, the changes will be in a file (named README) on diskette 1 of the configuration program diskettes. You can view the file with an ASCII text editor.

About the Software

IBM Nways Multiprotocol Access Services is the software that supports the IBM 2216 (licensed program number 5765-B87). This software has these components:

- The base code, which consists of:

  - The code that provides the routing, bridging, data link switching, and SNMP agent functions for the device.

  - The router user interface, which allows you to configure, monitor, and use the IBM Nways Multiprotocol Access Services base code installed on the device. The router user interface is accessed locally through an ASCII terminal or emulator attached to the service port, or remotely through a Telnet session or modem-attached device.

  The base code is installed at the factory on the 2216.

- The Configuration Program for IBM Nways Multiprotocol Access Services (*Configuration Program*), a graphical user interface that allows you to configure

the device from a stand-alone workstation. The Configuration Program
includes error checking and online help information.

The Configuration Program is not preloaded at the factory; it is shipped
separately from the device as part of the software order.

You can also FTP the Configuration Program for IBM Nways Multiprotocol
Access Services. See *Configuration Program User's Guide for Nways
Multiprotocol Access Services Version 1 Release 1,* GC30-3830, for the server
address and directories.

Conventions Used in This Manual

The following conventions are used in this manual to show command syntax and
program responses:

1. The abbreviated form of a command is shown in the following example:

   <u>rel</u>oad

   In this example, you can enter either the whole command (reload) or its
   abbreviation (rel).

2. Three periods following an option mean that you enter additional data (for
   example, a variable) after the option. For example:

   <u>t</u>ime host ...

   In this example, you enter the IP address of the host in place of the periods, as
   explained in the description of the command.

3. In information displayed in response to a command, defaults for an option are
   enclosed in brackets immediately following the option. For example:

   ```
   Media (UTP/STP) [UTP]
   ```

   In this example, the media defaults to UTP unless you specify STP.

4. Keyboard key combinations are indicated in text in the following ways:

   Ctrl-P

   `Ctrl` `P`

Common Tasks and the IBM 2216 Library

Planning and Installation

Planning and Setup Guide
GA27-4105

Hardware Installation Guide
GA27-4106

Hardware Configuration Quick Reference Card
GX27-3988

Configuration

Configuration Program User's Guide

GC30-3830

Configuration Program READ.ME
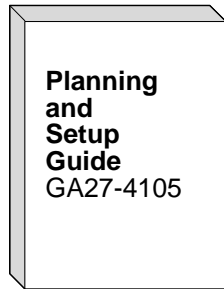
Configuration Help
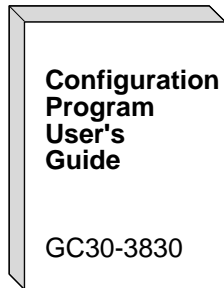
Diagnostics/ Maintenance

Maintenance Guide
SY27-0350
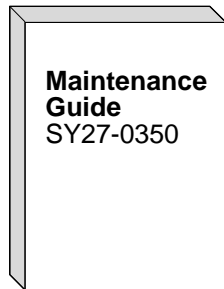
Operations and Network Administration

Software User's Guide
SC30-3886

Protocol Configuration and Monitoring Reference
SC30-3884
SC30-3885

Event Logging System Messages Guide
SC30-3682
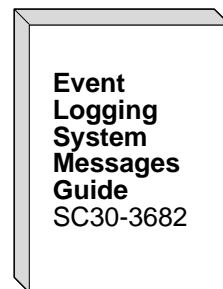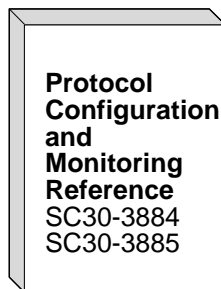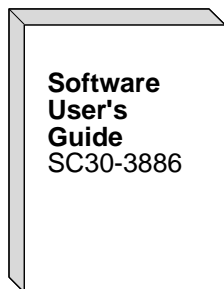
Library Overview

The following list shows the books in the IBM 2216 library, arranged according to tasks.

Information updates and corrections: To keep you informed of engineering changes, clarifications, and fixes that were implemented after the books were printed, refer to the IBM 2216 home pages at:

```
    http://www.networking.ibm.com/216/216prod.html
  and
    http://www.networking.ibm.com/216/216lib.html
```

Planning

GA27-4105   *IBM 2216 Multiaccess Connector Planning and Setup Guide*

       This book is shipped with the IBM 2216. It explains how to prepare for installation and perform an initial configuration.

Installation

GA27-4106   *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*

       This booklet is shipped with the IBM 2216. It explains how to install the IBM 2216 and verify its installation.

GX27-3988   *2216 Nways Multiaccess Connector Hardware Configuration Quick Reference*

       This reference card is used for entering and saving hardware configuration information used to determine the correct state of an IBM 2216.

Diagnostics and Maintenance

SY27-0350   *2216 Nways Multiaccess Connector Maintenance Guide.*

       This book is shipped with the IBM 2216 It provides instructions for diagnosing problems with and repairing the IBM 2216.

Operations and Network Management

The following list shows the books that support the Nways Multiprotocol Access Services program.

SC30-3886   *Nways Multiprotocol Access Services Software User's Guide*

       This book explains how to:

- Configure, monitor, and use the Nways Multiprotocol Access Services software.

- Use the Nways Multiprotocol Access Services command-line router user interface to configure and monitor the network interfaces and link-layer protocols shipped with the IBM 2216.

SC30-3884   *Nways Multiprotocol Access Services Protocol Configuration and Monitoring Reference, Volume 1*

SC30-3885   *Nways Multiprotocol Access Services Protocol Configuration and Monitoring Reference, Volume 2*

These books describe how to access and use the Nways Multiprotocol Access Services command-line user interface to configure and monitor the routing protocol software shipped with the product.

They include information about each of the protocols that the devices support.

SC30-3682   *Nways Event Logging System Messages Guide*

This book contains a listing of the error codes that can occur, along with descriptions and recommended actions to correct the errors.

Configuration

GC30-3830   *Configuration Program User's Guide*

This book discusses how to use the Nways Multiprotocol Access Services Configuration Program.

Safety

SD21-0030   *Caution: Safety Information—Read This First*

This book, shipped with the IBM 2216, provides translations of caution and danger notices applicable to the installation and maintenance of a IBM 2216.

Marketing

URL: **http://www.networking.ibm.com/216/216prod.html**

This IBM Web page provides product information through the World Wide Web.

Chapter 1. Getting Started (Introduction to the User Interface)

This chapter shows you how to get started with using the following components related to the IBM 2216 Model 400 Switch (2216) and the IBM Nways Multiprotocol Access Services:

- Router console terminals
- Router software (IBM Nways Multiprotocol Access Services)
- Router software user interface
- Protocol software - configuring and monitoring
- Network interfaces - configuring and monitoring

The information in this chapter is divided into the following sections:

- "Before You Begin"
- "Using Local and Remote Router Consoles"
- "Discussing the User Interface System" on page 1-7
- "Accessing Protocol Configuration and Console Processes" on page 1-10
- "Accessing Feature Configuration and Console Processes" on page 1-14
- "Accessing Network Interface Configuration and Console Processes" on page 1-15
- "System Security" on page 1-20

Before You Begin

Before you start, refer to the following checklist to verify that your router is installed correctly.

HAVE YOU...

- Installed all necessary hardware?

- Connected the console terminal (video terminal) to the router?

  Attention: If you are using a service port-attached terminal to configure or monitor your IBM 2216 and your service terminal is unreadable, you need to change some parameters in your configuration.

- Connected your router to the network using the proper network interfaces and cables?

- Run all necessary hardware diagnostics?

For more information on any of these procedures, refer to the *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*.

Using Local and Remote Router Consoles

The router console lets you use the router user interface to monitor and change the function of the router's networking software (IBM Nways Multiprotocol Access Services). The router supports local and remote consoles.

Local Consoles

Local consoles are either directly connected by an EIA 232 (RS-232) cable, or connected via modems to the router. You may need to use a local console during the initial software installation. After the initial setup connection, you can connect via Telnet, as long as IP forwarding has been enabled. (Refer to *Protocol Configuration and Monitoring Reference* for more information on enabling IP forwarding.)

When the configured router is started for the first time, a boot message appears on the screen, followed by the OPerator's CONsole or OPCON prompt (*). The * prompt indicates that the router is ready to accept OPCON commands. You will need to use an ASCII terminal attached to the 2216 service port to initially configure it.

Attention: Garbage, random characters, reverse question marks, or the inability to connect your terminal to the IBM 2216 service port can have many causes. The following list contains some of those causes:

- The most common cause of garbage or random characters on the service console is that the baud rate is not synchronized with the 2216.

  If the 2216 is set to a specific baud rate, the terminal or terminal emulator must be set to the same baud rate.

  If the 2216 is set to autobaud (this is the default), press the terminal `break` key sequence and press **Enter**.

  A typical break key sequence for PC terminal emulators is `Alt-b` (refer to the terminal emulator documentation). Most ASCII terminals have a `Break` key (often used in conjunction with the `Ctrl` key).

- Defective terminal or device (A/C) grounds.

- Defective, incorrectly shielded, or incorrectly grounded EIA 232 (RS-232) cable between the terminal and the IBM 2216.

- Defective terminal or terminal emulator.

- Defective IBM 2216 system board.

- High ambient electromagnetic interference (EMI) levels.

- Power line disturbances.

Once the 2216 is initially configured, you will not need a local console for router operation, as long as IP forwarding is enabled.

The router software automatically handles console activity. When upgrading the software, you might have to use the local console. For information on attaching and configuring local consoles, refer to the *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*.

Remote Consoles

Remote consoles attach to the router using a standard remote terminal protocol. Remote consoles provide the same function as local consoles, except that a local console must be used for initial configuration. You can use no more than two remote consoles at the same time on a router. You can connect remote consoles to the router via a Telnet connection (see "Telnet Connections"). You have the option to disable this feature.

Telnet Connections

The router supports both Telnet Client and Server. The remote console on the router acts as a Telnet server. The router acts as a Telnet client when connecting from the router to either another router or a host using the **telnet** command in the OPCON (*) process.

Remote Login Names and Passwords

During a remote login, the router prompts you for a login name and password. You can display the login name when logged in to the router from a remote console by using a router **status** command.

Use the **set password** command to supply a password for the router. The password, user-configurable for each router, controls access to the router. You may also configure a password for users of a local console.

Note: If you do not enter a login name and valid password within one minute of the initial prompt, or if you enter an incorrect password three times in succession, the router drops the Telnet connection.

Multiple users with login permissions may also be added using the **add user** command. See Chapter 3, "The CONFIG Process and Commands" for more details on the **set password** and **add user** commands.

Logging In Remotely or Locally

Logging in to a local console is the same as logging in to a remote console except that you must connect to the router by starting Telnet on your host system. To log in remotely, begin at step 1. To log in locally, begin at step 3.

To log in from a remote console:

1. Connect to the router by starting Telnet on your host system.

   Your host system is the system to which remote terminals are connected.

2. Supply the router's name or Internet Protocol (IP) address.

   To use router names, your network must have a name server. Issue either the router name or the IP address as shown in the following example:

   ```
   % telnet brandenburg
   ```

   or

   ```
   % telnet 128.185.132.43
   ```

   The router supports both Telnet Client and Server. The remote console on the router acts as a Telnet server. The router acts as a Telnet client when connecting from the router to either another router or a host using the **telnet** command.

   At this point, there is no difference whether you have logged in remotely or locally.

3. If you are prompted, enter your login name and password.

   ```
   login:
   Password:
   ```

   It is possible that there is a login and no password. The password controls access to the router. If a password has not been set, press the **Enter** key at the `Password:` prompt. Logins are not set automatically. For security, you can

set up user names and passwords using the **add user** command in the CONFIG process. For additional information, see the **add user** configuration command on page 3-8. Remember to reload or restart to activate their use.

Note: If you do not enter a login name and valid password within one minute of the initial prompt, or if you enter an incorrect password three times in succession, the router drops the Telnet connection.

4. Press the **Enter** key to display the main prompt asterisk (*).

You may have to press the **Enter** key more than once or press `Ctrl` `P` to obtain the * prompt.

Once at this level, you can begin to enter commands from the keyboard. Press the **backspace** key to delete the last character typed in on the command line. Press the **Delete** key or `Ctrl` `U` to delete the whole command line entry so that you can re-enter a command.

Once at the main router prompt, you can begin entering commands from the keyboard. (See "Discussing the User Interface System" on page 1-7 for more information on using the router user interface.) To exit the router, return to the main router prompt (*) and close the Telnet connection by typing **logout**. For example:

```
IP Config> exit
Config> Ctrl    P
* logout

%
```

You can also use local Telnet commands on your Telnet client to close the Telnet connection.

Executing a Command

When typing a command, remember the following:

- Type only enough sequential letters of the command to make it unique among available commands. For example, to execute the **reload** command you must enter **rel** as a minimum.

- Commands are not case-sensitive.

- Sometimes, only the first letter of the command (and subsequent options) is required to execute the command. For example, typing **s** at the * prompt followed by pressing the **Enter** key causes the **status** command be executed.

  Note: If you use a VT100 terminal, do not press the **Backspace** key, because it inserts invisible characters. Use the **Delete** key.

Connecting to a Process

When you start the router, the console displays a boot message. The OPCON prompt (*) then appears on the screen indicating that you are in the OPCON process and you can begin entering OPCON commands. This is the command prompt from which you communicate with different processes.

To connect your console to a process:

1. Find out the process ID (PID) number of a process by entering the **status** command at the * prompt.

The **status** command displays information about the router processes, such as the process IDs (PIDs), process names and status of the process. Issuing the **status** command is shown in the following example:

```
* status

Pid  Name     Status TTY  Comments
1    COpCon   RDY    TTY0
2    Monitr   DET    --
3    Tasker   RDY    --
4    MOSDDT   DET    --
5    CGWCon   DET    --
6    Config   DET    --
7    ROpCon   IDL    TTY1 128.185.210.125
8    ROpCon   IDL    TTY2
```

2. Use the **talk *pid*** command, where *pid* is the number of the process to which you want to connect. (For more information about these and other OPCON commands, refer to Chapter 2, "The OPCON Process and Commands.")

   Note: All the processes listed do not have a user interface (for example, the **talk 3** process). The **talk 4** command is for use by the software specialist.

Identifying Prompts

Each process uses a different prompt. You can tell which process your console is connected to by looking at the prompt. (If the prompt does not appear when you enter the **talk** *pid* command, press the **Return** key a few times.)

The following list shows the prompts for the three main processes:

Process	Prompt
OPCON	*
GWCON	+
CONFIG	Config>

At the prompt level, you can begin to enter commands from the keyboard. Use the **Backspace** key to delete the last character typed in on the command line. Use **Ctrl** **U** to delete the whole command line entry so that you can re-enter a command.

Getting Help

At any of the prompts just described, you can obtain help in the form of a listing of the commands available at that prompt level. This is done by typing **?** (the **help** command), and then pressing the **Return** key. Use **?** to list the commands that are available from the current prompt level. You can usually enter a **?** after a specific command name to list its options. For example, the following appears if you enter **?** at the * prompt:

```
*?
BREAKPOINT
DIVERT output from process
FLUSH output from process
HALT output from process
INTERCEPT character is
LOGOUT
MEMORY statistics
RELOAD
STATUS of process(es)
TALK to process
TELNET to IP-Address
```

Getting Back to OPCON

To get back to the OPCON prompt (*), press **Ctrl** **P** . You must always return to OPCON before you can communicate with another process. For example, if you are connected to the GWCON process and you want to connect to the CONFIG process, you must press **Ctrl** **P** to return to OPCON first. The **Ctrl** **P** key combination is called the *intercept character*.

When using third-level processes, such as IP Config or IP shown in Figure 1-1, use the **exit** command to return to the second level.

Use the intercept character (the default intercept character is **Ctrl** **P**) from a third-level process to return to the * prompt. The next time you use the **talk** command, you will re-enter the third level process. This link goes away when the router is reinitialized.

```
First                              ┌────────┐
Level                        ┌────▶│ OPCON  │◀────────────┐
                             │     │   *    │             │
                             │     └────────┘             │
                             │      │    ▲                │
              Talk 5         │      │    │    Talk 6       │
                             │      ▼    │    ▼            │
                             │  ┌────────┐   ┌────────┐    │
Second                       │  │ GWCON  │ctrl-p│CONFIG │  │
Level                        │  │process │──────│process│  │
                             │  └────────┘   └────────┘    │
                             │    │   ▲         │   ▲       │
               Prot 0        │    │   │         │   │  Prot 0
                             │    ▼   │         ▼   │       │
       ctrl-p                │  ┌────────┐   ┌────────┐ ctrl-p
Third                        └──┤ IP> │Exit│ Exit│IP Config>├──┘
Level                           └────────┘   └────────┘
```
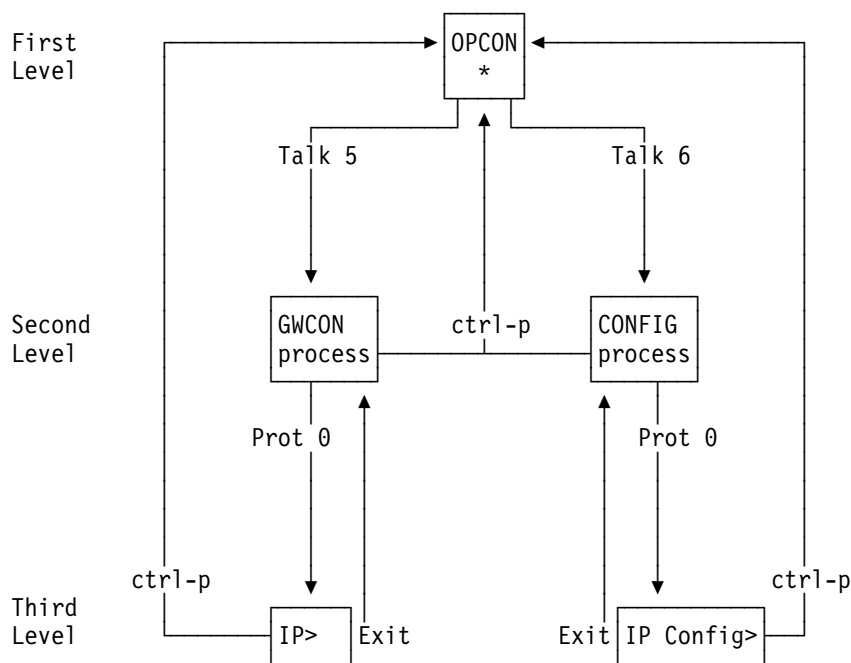
Figure 1-1. Talk and the Intercept Commands

Figure 2-2 on page 2-3 shows the commands in the OPCON structure.

Exiting the Router

Return to the * prompt and close the Telnet connection. For example:

```
IP Config> exit
Config> Ctrl  P
* logout

%
```

You can also use local Telnet commands on your Telnet client to close the Telnet connection.

Discussing the User Interface System

The user interface to the router software consists of the main menu (process) and several subsidiary menus (processes). The processes you use most often are OPCON, GWCON, CONFIG, and CONFIG-ONLY. These processes allow you to control and monitor the operations of the router.

There are three levels of processes: first, second, and third as you move down the software tree.

Figure 1-2 shows the processes and how they fit within the structure of the router software.

Router Software Processes
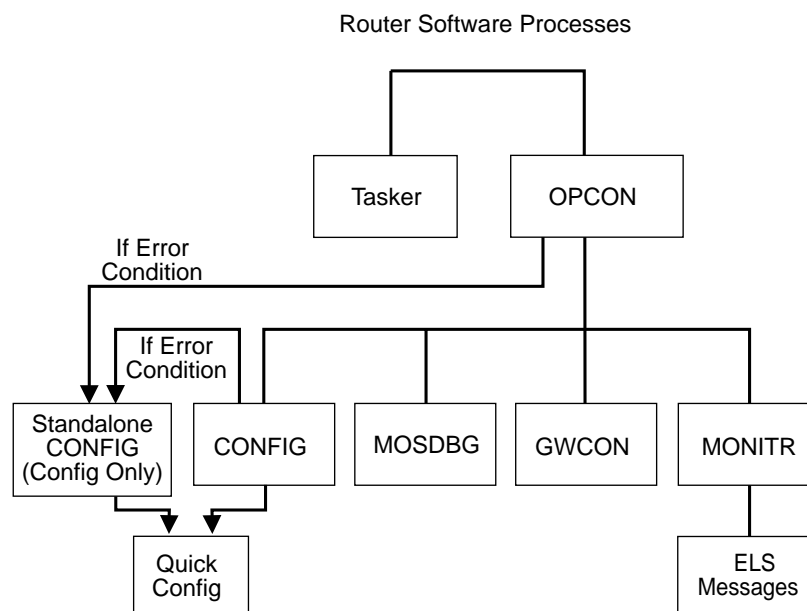
Figure 1-2. IBM Nways Multiprotocol Access Services

The router software (IBM Nways Multiprotocol Access Services) is a multi-tasking system that schedules use of the CPU among various processes and hardware devices. The router software:

- Provides timing and memory management, and supports both local and remote operator consoles from which you can view and modify the router's operational parameters.

- Consists of functional modules that include various user interface processes, all network interface drivers, and all protocol forwarders purchased with the router.

Table 1-1 on page 1-8 summarizes the IBM Nways Multiprotocol Access Services processes and functions on the 2216.

Table 1-1. Router Software Processes

Process	Definition	Prompt
OPCON	Operates as the main operator control program. It provides service for one directly connected console terminal.	Asterisk (*)
ROPCON	Provides OPCON service for two remotely connected console terminals (not shown). Functionally, OPCON and ROPCON are the same.	Asterisk (*)
GWCON (or CGWCON) (Second Level)	Displays the status and statistics of the router's hardware and software, such as protocols, network interfaces, and event logging. It is on the second level of the router user interface. It provides access to the third-level process, which allow you to monitor configured protocols and features.	Plus sign (+)
CONFIG (Second Level)	Provides online control of various configuration parameters, such as network addresses and event logging. It is on the second level of the router user interface. It provides access to third-level processes, which allow you to configure various protocols and features. Quick Config menus are also available under this process.	`Config>`
MONITR	Receives Event Logging System (ELS) messages and messages from the operating system, and displays then on the monitor, according to user-selected filtering criteria. It is on the second level of the router user interface.	None
TASKER	Runs the router's main networking software and performs the router's internetwork data transfer operations. This process is part of the operating system and has no user interface.	None
MOSDDT	Serves as the Micro Operating System (MOS) runtime debugging tool (Dynamic Debugging Tool).	Dollar sign ($)
QUICK CONFIG	Provides a simple, less-detailed way of configuring devices, bridging and routing protocols, and booting records.	None
CONFIG-ONLY (Standalone Config)	Provides the same function as the CONFIG process with the addition of the **reload** command.	`Config (only)>`

Definition of the First-Level User Interface

The first level of the user interface includes CONFIG-ONLY and OPCON. The CONFIG-ONLY process (which includes (Quick Configuration) allows you to configure the router. The OPCON process allows you to set up communication between users and the other router processes. If you are using a remote console, the name of the process handling your console is ROPCON instead of OPCON, but the operation is identical.

CONFIG-ONLY Process

Configuration Only, or CONFIG-ONLY, allows you to take the router off line and reconfigure its operating parameters.

The commands available are the same as CONFIG at the `Config>` prompt; however, no other processes are running while the router is in CONFIG-ONLY mode.

The CONFIG-ONLY mode uses commands identical to the CONFIG process, with the addition of the **reload** command.

CONFIG-ONLY mode is provided only for getting a subset of configuration commands when a configuration problem causes the router to crash with a *panic*, *check*, *fatal*, or *bughlt*, particularly those relating to failures of memory allocations. CONFIG-ONLY mode should be used only to adjust parameters affecting memory allocations such as routing table sizes, packet sizes, and receive buffer allocations. It should not be used for general router configuration. In CONFIG-ONLY mode, many of the device-related commands are disabled and some may cause a crash.

There are two ways to enter CONFIG-ONLY mode:

Operational Failure: The router encounters a problem during initialization and automatically comes up in CONFIG-ONLY mode. Any of the following situations will cause the router to enter into CONFIG-ONLY mode:

- An unsupported device is in the software load

- One or more of the following configuration errors have occurred during startup:
    - A configuration file on the hard drive is corrupted.
    - All router interface information has been deleted.
    - Incorrect interface configuration information has been entered.

Deliberate entry into CONFIG-ONLY mode: To access CONFIG-ONLY mode deliberately, select a configuration bank that has a file that does not contain a configuration. Files that do not contain configurations are indicated by *NONE* in the output of the **list configuration** command. See "CONFIG-Only Mode" on page 3-2 for additional details about entering CONFIG-ONLY mode.

Quick Configuration Process

Quick Configuration, or Quick Config, allows you to quickly configure portions of the router without dealing with the specific operating system commands. When you initially load, start, or restart the router with no configuration, you enter Config-Only and you can access Quick Config menus from that process. If the router has devices configured and the devices do not have any protocols configured, the router automatically starts Config-Only and then enters Quick Config.

You can also enter Quick Config from the CONFIG process using the **qconfig** command.

OPCON

The OPCON process handles the communication between users and the other router processes. If you are using a remote console, the name of the process handling your console is ROPCON instead of OPCON, but the operation is identical.

Refer to Chapter 2, "The OPCON Process and Commands" for complete details.

Accessing Protocol Configuration and Console Processes

To help you access the configuration and console processes, this section outlines both of these procedures.

All protocols described in the *Protocol Configuration and Monitoring Reference* have commands that are executed in one of the following ways:

- Accessing the protocol *configuration* process to initially configure and enable the protocol, as well as perform later configuration changes.

- Accessing the protocol *console* process to monitor information about each protocol or make temporary configuration changes.

The procedure for accessing these processes is basically the same for all protocols. The next sections describe these procedures.

Accessing the Protocol Configuration Process (CONFIG)

Each protocol configuration process is accessed through the router's CONFIG process. CONFIG is the second-level process of the router user interface that lets you communicate with third-level processes. Protocol processes are examples of third-level processes.

The CONFIG command interface is made up of levels that are called modes. Protocol configuration command interfaces are modes of the CONFIG interface. Each protocol configuration interface has its own prompt. For example, the prompt for the TCP/IP protocol command interface is IP config>.

To access the protocol configuration processes:

1. Enter the CONFIG command process from OPCON and obtain the CONFIG prompt (Config>)

2. Enter the desired protocol configuration process (with its own prompt) from the CONFIG prompt using the **protocol** command.

The next sections describe these procedures in more detail.

Entering the CONFIG Process

To enter the CONFIG command process from OPCON and obtain the CONFIG prompt:

1. At the OPCON prompt, enter the **status** command to find the PID for CONFIG. (See page 1-5 for sample output of the **status** command.)

2. Enter the OPCON **talk** command and the PID for CONFIG. The PID for CONFIG is 6.

   ```
   * talk 6
   ```

The console displays the CONFIG prompt (Config>). If the prompt does not appear, press the **Return** key again.

Quick Configuration Process Quick Configuration, or Quick Config, allows you to quickly configure portions of the router without dealing with the specific operating system commands. You enter the Quick Config menus from the CONFIG process using the **qconfig** command.

Entering the Desired Protocol Configuration Process

To enter the desired protocol configuration process from the CONFIG prompt:

1. At the CONFIG prompt, enter the **list configuration** command to see the numbers and names of the protocols purchased in your copy of the software. See 3-24 for sample output of the **list configuration** command.

2. From the Config> prompt, enter the **protocol** command with the number or short name (for example, IP, IPX, and ARP) of the protocol you want to configure. The protocol number and short name is obtained from the **list configuration** command display. In the following example, the command has been entered for accessing the IP protocol configuration process:

   Config> **protocol IP**

     or

   Config> **protocol 0**

   The protocol configuration prompt then displays on the console. The following example shows the IP protocol configuration prompt:

   IP config>

You can now begin entering the protocol's configuration commands. See the corresponding protocol section of the *Protocol Configuration and Monitoring Reference* for more information on specific protocol configuration commands.

In summary, the **protocol** command lets you enter the configuration process for the protocol software installed in your router. The **protocol** command enters a protocol's command process. After entering the protocol command, the prompt of the specified protocol appears. From the prompt, you can enter commands specific to that protocol.

Exiting the Protocol Configuration Process

To exit the protocol configuration process:

1. Return to the CONFIG process by entering the protocol **exit** command. For example:

   IP config> **exit**

2. Return to the OPCON process by entering the OPCON intercept character (**Ctrl** **P**). For example:

   Config> **Ctrl** **P**

Reloading the Router

Changes that you make to the protocol parameters through CONFIG do not take effect until you reload the router software.

Note: The changes you make through CONFIG are retained in a configuration database in nonvolatile memory. They are retained during power downs and are recalled when you restart the router.

To reload the router, enter the OPCON **reload** command. For example:

```
* reload
```

```
Are you sure you want to reload the gateway? (Yes or No):  yes
```

Accessing the Protocol Console (Monitoring) Process, GWCON

To view information about the protocols or to change parameters at the console, you must access and use the protocol console process. Protocol console command interfaces are modes of the GWCON interface. Within the GWCON mode, each protocol console interface has its own prompt. For example, the prompt for the TCP/IP protocol is IP>.

To access the protocol console processes:

1. Enter the GWCON command process from OPCON and obtain the GWCON prompt.

2. Enter the desired protocol console process from the GWCON (+) prompt using the **protocol** command.

   For more information on using the **protocol** command, see "Protocol" on page 6-12.

The next sections describe these procedures in more detail.

Entering the GWCON Command Process

To enter the GWCON process from OPCON and obtain the GWCON prompt:

1. At the OPCON (*) prompt, enter the **status** command to find the PID for GWCON. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter the OPCON **talk** command and the PID for GWCON. For example:

   ```
   * talk 5
   ```

   The GWCON prompt (+) then displays on the console. If the prompt does not appear, press **Return** again.

Entering a Protocol Console Process

To enter a protocol console process from the GWCON prompt:

1. At the GWCON prompt, enter the **configuration** command to see the protocols and networks configured for the router. For example: **+configuration**

```
Multiprotocol Access Services

5765-B87 Feature 2802 V1 R1.0 PTF 0 RPQ 0



Num Name  Protocol
0   IP    DOD-IP
3   ARP   Address Resolution
4   DN    DNA Phase IV
6   VIN   Banyan Vines
7   IPX   NetWare IPX
10  BGP   Border Gateway Protocol
11  SNMP  Simple Network Management Protocol
12  OSPF  Open SPF-Based Routing Protocol
22  AP2   AppleTalk Phase 2
23  ASRT  Adaptive Source Routing Transparent Enhanced Bridge
26  DLS   Data Link Switching
27  XTP   X.25 Transport Protocol
28  APPN  Advanced Peer-to-Peer Networking [HPR]
30  APPN  Advanced Peer-to-Peer Networking [ISR]

Num Name  Feature
2   MCF   MAC Filtering

16 Networks:
Net Interface  MAC/Data-Link        Hardware             State
0   TKR/0      Token-Ring/802.5     Token-Ring           Up
1   TKR/1      Token-Ring/802.5     Token-Ring           Up
2   TKR/2      Token-Ring/802.5     Token-Ring           Up
3   TKR/3      Token-Ring/802.5     Token-Ring           Up
4   Eth/0      Ethernet/IEEE 802.3  Ethernet             Up
5   Eth/1      Ethernet/IEEE 802.3  Ethernet             Up
6   Eth/2      Ethernet/IEEE 802.3  Ethernet             Up
7   Eth/3      Ethernet/IEEE 802.3  Ethernet             Up
8   Eth/4      Ethernet/IEEE 802.3  Ethernet             Up
9   Eth/5      Ethernet/IEEE 802.3  Ethernet             Up
10  FR/0       Frame Relay          V.35/V.36            Up
11  X25/0      X.25                 V.35/V.36            Up
12  PPP/0      Point to Point       V.35/V.36            Up
13  PPP/1      Point to Point       V.35/V.36            Up
14  PPP/2      Point to Point       V.35/V.36            Up
15  PPP/3      Point to Point       V.35/V.36            Up
```

2. Enter the GWCON **protocol** command with the protocol number or short name of the desired protocol displayed in the configuration information.

   In the following example, the command has been entered for accessing the IP protocol console process.

   + **protocol 0**

   or

   + **protocol IP**

   The protocol console prompt then displays on the console. This example shows the IP protocol console prompt:

   ```
   IP>
   ```

You can now begin entering the protocol's commands. See the corresponding protocol section of the *Protocol Configuration and Monitoring Reference* for more information on specific protocol console commands.

Exiting the Protocol Console Process

To exit the protocol console process and return to the OPCON process:

1. Return to the GWCON process by entering the protocol **exit** command. For example:

   IP> **exit**

2. Return to the OPCON process by entering the OPCON intercept character (**Ctrl** **P**). For example:

   + **Ctrl** **P**

Protocol Names and Numbers

Table 1-2 lists the numbers that you enter with the **protocol** command to access a specific protocol configuration or console process.

Table 1-2. Protocol Numbers and Names		
Protocol Number	**Protocol Short Name**	**Accesses the following protocol process**
0	IP	IP (Internet Protocol)
3	ARP	ARP (Address Resolution Protocol)
4	DN	DNA Phase IV
6	VIN	Banyan VINES
7	IPX	IPX (Novell NetWare Internetwork Packet Exchange)
8	OSI	ISO CLNP/ESIS/ISIS
9	DVM	Distance Vector Multicast Routing Protocol
10	BGP	BGP (Border Gateway Protocol)
11	SNMP	SNMP (Simple Network Management Protocol)
12	OSPF	OSPF (Open Shortest Path First)
20	SDLC	SDLC Relay
22	AP2	AppleTalk Phase 2
23	ASRT	Adaptive Source Routing Transparent Bridge
24	HST	TCP/IP Host Services
25	LNM	LAN Network Manager
26	DLS	Data Link Switching
27	XTP	X.25 Transport Protocol
28	APPN HPR	APPN High Performance Routing
30	APPN ISR	APPN Intermediate Session Routing

Accessing Feature Configuration and Console Processes

To help you access the IBM Nways Multiprotocol Access Services feature configuration and console processes, this section outlines both of these procedures.

All IBM Nways Multiprotocol Access Services features described in this guide have commands that are executed in one of the following ways:

- Accessing the *configuration* process to initially configure and enable the feature as well as perform later configuration changes.
- Accessing the *console* process to monitor information about each feature or make temporary configuration changes.

The procedure for accessing these processes is basically the same for all features. The next sections describe these procedures.

Accessing the Feature Processes

Use the **feature** command from the CONFIG process to access configuration commands for specific IBM Nways Multiprotocol Access Services features outside of the protocol and network interface configuration processes.

Use the **feature** command from the GWCON process to access console commands for specific features that are outside of the protocol and network interface console processes.

Enter a question mark after the **feature** command to display a listing of the features available for your software release. For example:

```
Config> feature ?

WRS
BRS
MCF

Feature name or number [1] ?
```

To access a particular feature's configuration or console prompt, enter the **feature** command at the Config> or + (GWCON) prompt, respectively, followed by the feature number or short name. For example:

```
Config> feature mcf

MAC filtering user configuration

Filter Config>
```

Table 3-4 on page 3-23 lists the available feature numbers and names.

Once you access the configuration or console prompt for a feature, you can begin entering specific commands for the feature. To return to the previous prompt level, enter the **exit** command at the feature's prompt.

Accessing Network Interface Configuration and Console Processes

This section describes how to get started with accessing the network interface configuration and console processes. Accessing these processes lets you change and monitor software-configurable parameters for network interfaces used in your router.

Although this manual concerns itself primarily with configuration tasks, there might be some point during configuration when you want to display some interface statistics and information. To help you access both the configuration *and* console processes, this manual outlines both procedures. The information presented in the next sections includes:

- Accessing the Network Interface Configuration Process

• Accessing the Network Interface Console Process

Accessing the Network Interface Configuration Process

Use the following procedure to access the router's configuration process. This process gives you access to a specific interface's *configuration* process.

1. At the OPCON prompt (*), enter the **status** command to find the PID for CONFIG. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter the OPCON **talk** command and the PID for CONFIG. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

```
* talk 6
```

After you enter the talk 6 command, the CONFIG prompt (Config>) displays on the console. If the prompt does not appear when you first enter **CONFIG**, press **Return** again.

Use the **add device** command to create a network interface. The add device command automatically assigns the interface number and supports the following types of devices:

a. Multi-port adapters

When you specify a multi-port adapter device name with the **add device** command, you are prompted for the adapter's slot number and the port number on the adapter that you want to use for the interface.

If you want to use multiple ports on an adapter, you must enter the add device command multiple times and specify a different port number each time.

For example, you would enter the following commands to create interfaces for ports 0 and 1 on the 8-port X.21 adapter in slot 7.

```
Config> add device x21
Device Slot #(1-8) [1]? 7
Device Port #(0-7) [0]? 0
Defaulting Data-link protocol to PPP
Adding X.21 PPP device in slot 7 port 0 as interface #6
Use "set data-link" command to change the data-link protocol
Use "net 6" to configure X.21 PPP parameters

Config> add device x21
Device Slot #(1-8) [1]? 7
Device Port #(0-7) [0]? 1
Defaulting Data-Link protocol to PPP
Adding X.21 PPP device in slot 7 port 1 as interface #7
Use "set data-link" command to change the data-link protocol
Use "net 7" to configure X.21 PPP parameters
```

Note: The serial adapter port numbers are 0-based. The port numbers for all other multi-port adapters are 1-based.

b. Single-port adapters

When you specify a single-port adapter device name with the **add device** command, you are prompted for the adapter's slot number.

The following example adds an interface for the 1-port ISDN-PRI T1/J1 adapter in slot 2.

```
Config> add device t1-isdn
Device Slot #(1-8) [1]? 2
Adding ISDN Primary T1/J1 device in slot 2 port 1 as interface #7
Use "net 7" to configure ISDN Primary T1/J1 parameters
```

c. Dial circuits

The following example adds a dial circuit interface:

```
Config> add device dial-circuit
Adding device as interface 8
Defaulting Data-link protocol to PPP
Use "set data-link" command to change the data-link protocol
Use "net 8" command to configure circuit parameters
```

Enter **add device ?** to get a list of the supported device types.

Note: When you create interfaces for serial adapters or dial circuits, the default data-link type is PPP. However, you can use the **set data-link** command to change the data-link type. Refer to Table 1-3 on page 1-18 for the data-link types supported on serial adapters and dial circuits, and to the description of the **set data-link** command on page 3-29.

IBM 2216 Device Support Restrictions:

The following rules apply when adding devices to the IBM 2216:

- As many as two ATM interfaces may be defined.

- As many as four ISDN-PRI interfaces may be defined.

- Adding a LAN device in slot 3, 4, 7, or 8 will disable the other slot in the pair. For example, if you add a LAN device to slot 4, slot 3 will be disabled. Likewise, if you add the device to slot 3, slot 4 will be disabled. The same rules apply for adding a LAN device in slots 7 and 8.

- All interfaces on a V.35/V.36 adapter must use either V.35 cables or V.36 cables. The type of fanout cable attached to the V.35/V.36 adapter determines which type of cables (V.35 or V.36) can be used.

3. At the Config > prompt, enter the **list devices** command to display the network interface numbers for which the router is currently configured, as follows:

```
Config> list devices
Ifc 0 Token Ring                    Slot: 1  Port: 1
Ifc 1 Token Ring                    Slot: 1  Port: 2
Ifc 2 Token Ring                    Slot: 2  Port: 1
Ifc 3 Token Ring                    Slot: 2  Port: 2
Ifc 4 Ethernet                      Slot: 4  Port: 1
Ifc 5 Ethernet                      Slot: 4  Port: 2
Ifc 6 Ethernet                      Slot: 5  Port: 1
Ifc 7 Ethernet                      Slot: 5  Port: 2
Ifc 8 Ethernet                      Slot: 6  Port: 1
Ifc 9 Ethernet                      Slot: 6  Port: 2
Ifc 10 V.35/V.36 Frame Relay        Slot: 8  Port: 0
Ifc 11 V.35/V.36 X.25               Slot: 8  Port: 1
Ifc 12 V.35/V.36 PPP                Slot: 8  Port: 2
Ifc 13 V.35/V.36 PPP                Slot: 8  Port: 3
Ifc 14 V.35/V.36 PPP                Slot: 8  Port: 4
Ifc 15 V.35/V.36 PPP                Slot: 8  Port: 5
```

4. Record the interface numbers.

5. Enter the CONFIG **network** command and the number of the interface you
   want to configure. For example:

   ```
   Config> network 1
   ```

   The appropriate configuration prompt (such as `TKR Config>` for token-ring), now
   displays on the console.

Note: Not all network interfaces are user-configurable. For interfaces that cannot
be configured, you receive the message:

```
That network is not configurable
```

Displaying the Interface Configuration

From the same interface configuration prompts, you can list configuration
information specific to that selected interface by using the **list** command. For
example:

```
TKR Config> list

Token-Ring configuration:

PACKET SIZE (INFO FIELD): 4472
Speed:                    16 Mb/sec
Media:                    Shielded

RIF Aging Timer:          120
Source Routing:           Enabled
MAC Address:              000000000000
```

Configuring the Network Interface

Refer to the specific chapters in this guide for complete information on configuring
your 2216's network interfaces.

Table 1-3 lists network architectures and the supported interfaces for each
architecture.

Table 1-3. Network Architecture and the Supported Interfaces	
Network Architecture	**Supported Interfaces**
ATM	1-Port ATM 155-Mbps MMF 1-Port ATM 155-Mbps SMF
802.5 Token-Ring	2-Port Token-Ring
Ethernet	2-Port Ethernet
ISDN	1-Port ISDN-PRI (T1/J1) 1-Port ISDN-PRI (E1)
Point-to-Point	8-port V.24/EIA 232E, 6-port V.35/V.36, 8-port X.21, and dial circuit interfaces
Frame Relay	8-port V.24/EIA 232E, 6-port V.35/V.36, 8-port X.21, and dial circuit interfaces
X.25	8-port V.24/EIA 232E, 6-port V.35/V.36, and 8-port X.21
SDLC Relay	8-port V.24/EIA 232E, 6-port V.35/V.36, and 8-port X.21
SDLC	8-port V.24/EIA 232E, 6-port V.35/V.36, and 8-port X.21
V.25 bis	8-port V.24/EIA 232E

Exiting the Interface Configuration Process

After you have configured the interface information, exit the interface configuration process by using the following procedure:

1. Return to the CONFIG process by entering the **exit** command. For example:

       TKR Config> **exit**

2. Return to the OPCON process by entering the OPCON intercept character. The default intercept character is **Ctrl** **P** . (See "Intercept" on page 2-6 for more information.) For example:

   Config> **Ctrl** **P**

Reloading the Router

Whenever you change a user-configurable parameter, you must reload the router for the change to take effect. To do so, enter the OPCON **reload** command. For example:

    * **reload**

    Are you sure you want to reload the gateway? (Yes or No): **yes**

Accessing the Network Interface Console Process

To monitor information related to a specific interface, access the interface console process by using the following procedure:

1. Enter the **status** command to find the PID for GWCON. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter the OPCON **talk** command and the PID for GWCON. For example:

       * **talk 5**

3. The GWCON prompt (+) is displayed on the console. If the prompt does not appear when you first enter GWCON, press **Return** again.

4. At the GWCON prompt, enter the **configuration** command to see the protocols and networks for which the router is configured. For example:

       +**configuration**
       Multiprotocol Access Services

       5765-B87 Feature 2802 V1 R1.0 PTF 0 RPQ 0

       Num Name Protocol
       0   IP   DOD-IP
       3   ARP   Address Resolution
       4   DN   DNA Phase IV
       6   VIN   Banyan Vines
       7   IPX   NetWare IPX
       10 BGP   Border Gateway Protocol
       11 SNMP Simple Network Management Protocol
       12 OSPF Open SPF-Based Routing Protocol
       22 AP2   AppleTalk Phase 2
       23 ASRT Adaptive Source Routing Transparent Enhanced Bridge
       26 DLS   Data Link Switching
       27 XTP   X.25 Transport Protocol
       28 APPN Advanced Peer-to-Peer Networking [HPR]
       30 APPN Advanced Peer-to-Peer Networking [ISR]

       Num Name Feature
       2   MCF   MAC Filtering

```
16 Networks:
Net Interface  MAC/Data-Link       Hardware           State
0   TKR/0      Token-Ring/802.5    Token-Ring         Up
1   TKR/1      Token-Ring/802.5    Token-Ring         Up
2   TKR/2      Token-Ring/802.5    Token-Ring         Up
3   TKR/3      Token-Ring/802.5    Token-Ring         Up
4   Eth/0      Ethernet/IEEE 802.3 Ethernet           Up
5   Eth/1      Ethernet/IEEE 802.3 Ethernet           Up
6   Eth/2      Ethernet/IEEE 802.3 Ethernet           Up
7   Eth/3      Ethernet/IEEE 802.3 Ethernet           Up
8   Eth/4      Ethernet/IEEE 802.3 Ethernet           Up
9   Eth/5      Ethernet/IEEE 802.3 Ethernet           Up
10  FR/0       Frame Relay         V.35/V.36          Up
11  X25/0      X.25                V.35/V.36          Up
12  PPP/0      Point to Point      V.35/V.36          Up
13  PPP/1      Point to Point      V.35/V.36          Up
14  PPP/2      Point to Point      V.35/V.36          Up
15  PPP/3      Point to Point      V.35/V.36          Up
```

5. Enter the GWCON **network** command and the number of the interface you want to monitor. For example:

   `+ `**`network 11`**

   In this example, the X.25 console prompt is displayed on the console. You can then view information about the X.25 interface by entering the X.25 console commands.

   `X.25>`

Monitoring the Network Interface

Refer to the specific chapters in this manual for complete information on monitoring your 2216's network interfaces.

Exiting the Interface Console Process

To exit the interface console process and return to the OPCON process:

1. Return to the GWCON process by entering the **exit** command. For example:

   `X.25> `**`exit`**

2. Return to the OPCON process by entering the OPCON intercept character. (The default intercept character is **Ctrl** **P** .) For example:

   `+ ` **Ctrl** **P**

System Security

Multiple users with login permissions can also be added using the **add user** command. See "Configuring User Access" on page 3-5 for details on security issues and for information on the **set password** and **add user** commands.

Chapter 2. The OPCON Process and Commands

This chapter describes the OPCON process and includes the following sections:

- "What is OPCON?"
- "Accessing the OPCON Process" on page 2-2
- "OPCON Commands" on page 2-3

What is OPCON?

The Operator Console process (OPCON) is the root-level process of the router software user interface. The main function of OPCON is to control which processes are connected to consoles. OPCON fits into the router software structure as shown in Figure 2-1. Using OPCON commands, you can:

- Manipulate the output from a process
- Change the intercept character
- Display information about router memory usage
- Reload the router software (reboot)
- Telnet to other routers or hosts
- Display status information about all router processes
- Communicate with processes at the secondary level
- Escape to the MOS system debugging tool

Router Software Processes

```
                         OPCON
                           *

   talk 6          talk 5                    talk 2

  config         GWCON        MOSDBG        MONITR
  CONFIG >         +

                                          ELS Messages
```
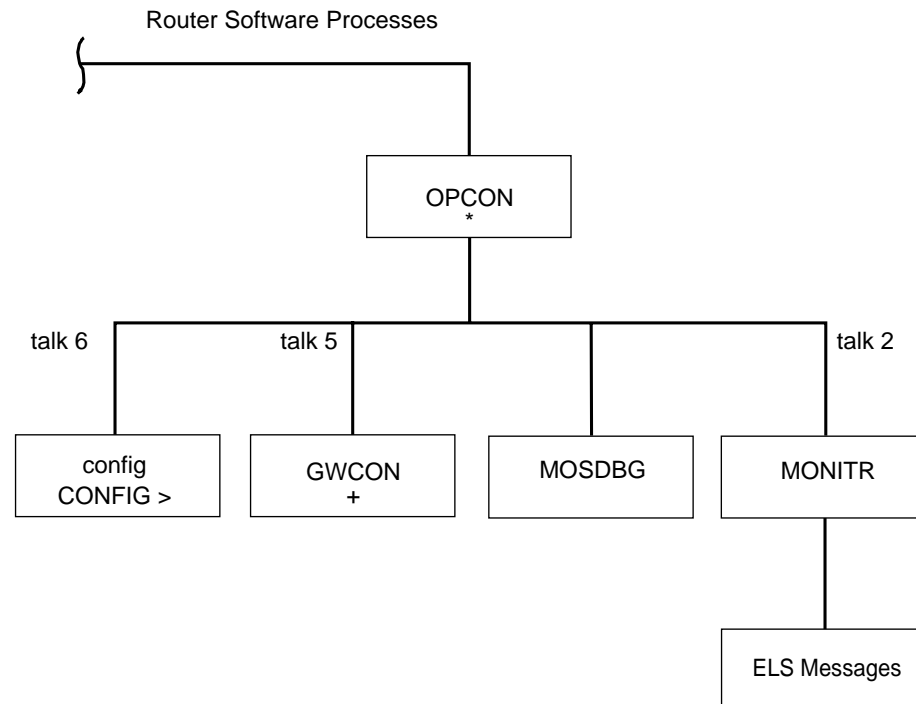
Figure   2-1. OPCON in the Router Software Structure

Accessing the OPCON Process

When you start the router for the first time, a boot message appears on the console. Then the OPCON prompt (*) appears on the console, indicating that you are in the OPCON process and can begin entering OPCON commands.

When you access OPCON from a remote terminal, you access the ROPCON (Remote Operator Console) process. ROPCON and OPCON are functionally the same.

The OPCON process allows you to configure and monitor all of the router's operating parameters. While in the OPCON process, the router is forwarding data traffic. When the router is booted and enters OPCON, a copyright logo and an asterisk (*) prompt appears on the locally attached console terminal. This is the OPCON (OPerator's CONsole) prompt, the main user interface that allows access to second-level processes.

Some changes to the router's operating parameters made while in OPCON take effect immediately without requiring reinitializing of the router. If the changes do not take effect, use the reload command at the * prompt.

At the * prompt, there is an extensive set of commands that you can enter to check the status of various internal software processes, monitor the performance of the router's interfaces and packet forwarders, and configure various operational parameters.

OPCON Commands

This section describes the OPCON commands. Each command includes a description, syntax requirements, and an example. The OPCON commands are summarized in Table 2-1 on page 2-4. To use them, access the OPCON process and enter the appropriate command at the OPCON prompt (*).

Figure 2-2 shows the OPCON Command Tree.

```
                        ┌──────────────┐
                        │ OPCON/ROPCON │
                        │      *       │
                        └──────────────┘
                             │        └────────────────TALK──────────────┐
                             │                         │ │ │             │
                        5    │                              6            2
                    ┌──────────┐              ┌──────────┐      ┌──────────┐
                    │ CGWCON   │              │ CONFIG   │      │ MONITR   │
                    │   +      │              │ Config>  │      │          │
                    └──────────┘              └──────────┘      └──────────┘
         HELP (?) ─┤           ├─ HELP (?)           ├─ HELP (?)
           DIAGS ──┤           ├─ BOOT information   ├─ ADD
hardware diagnostics│          ├─ BUFFER  statistics ├─ BOOT and load file
          DIVERT ──┤           ├─ CLEAR   statistics ├─ CHANGE
output of process   │          ├─ CONFIGURATION of gateway ├─ CLEAR
           FLUSH ──┤           ├─ DISABLE  interface ├─ DELETE
output of process   │          ├─ ERROR   counts     ├─ DISABLE
            HALT ──┤           ├─ EVENT  logging     ├─ ENABLE
output of process   │          ├─ FEATURE commands   ├─ EVENT
       INTERCEPT ──┤           ├─ INTERFACE  statistics ├─ FEATURE
       character    │          ├─ MEMORY statistics  ├─ LIST
          LOGOUT ──┤           ├─ NETWORK commands   ├─ NETWORK
          MEMORY ──┤           ├─ PROTOCOL commands  ├─ PATCH
      statistics    │          ├─ QUEUE lengths      ├─ PROTOCOL
          RELOAD ──┤           ├─ STATISTICS of network ├─ QCONFIG
          STATUS ──┤           ├─ TEST Network       ├─ SET
  of process(es)    │          └─ UPTIME of gateway  ├─ SYSTEM memory dump
            TALK ──┤                                 ├─ TIME of day parms
      to process    │                                ├─ UNPATCH
      TELNET to ────┤                                ├─ UPDATE
      IP-address                                     └─ WRITE the configuration
<this terminal type>
```

Figure   2-2. OPCON Command Tree

Table 2-1. OPCON Commands

Command	Function
? (Help)	Lists all the OPCON commands.
Diags	Displays device status and the contents of the hardware test log and the hardware error log.
Divert	Sends the output from a process to a console or other terminal.
Flush	Discards the output from a process.
Halt	Suspends the output from a process.
Intercept	Sets the OPCON default intercept character.
Logout	Logs out a remote console.
Memory	Reports the router's memory usage.
Reload	Reloads the router software.
Status	Shows information about all router processes.
Talk	Connects to another router process and enables the use of its commands.
Telnet	Connects to another router.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

```
DIAGS diagnostics
DIVERT output from process
FLUSH output from process
HALT output from process
INTERCEPT character is
LOGOUT
MEMORY statistics
RELOAD
STATUS of process(es)
TALK to process
TELNET to IP Address (this terminal type)
```

Diags

Use the **Diags** command to display the Diagnostic Main Menu. The diagnostic menus allow you to enable, disable and test hardware adapters or ports. Diagnostic menus have on-screen help for the various options and status information that is available.

You can use the "b" (back) key to return to any previous menu. Use the "e" (exit) key to exit the diagnostics and return to the OPCON command prompt.

Syntax: <u>diag</u>s

Example: diags

Divert

Use the **divert** command to send the output from a specified process to a specified terminal. This command allows you to divert the output of several processes to the same terminal to simultaneously view the output. The **divert** command is commonly used to redirect MONITR output messages to a specific terminal. The router allows only certain processes to be redirected.

After entering the command, enter the PID and tty# (number of the output terminal). To obtain these values, use the OPCON status command. The terminal number can be the number of either the local console (tty0) or one of the remote consoles (tty1, tty2). The following example shows Event Logging System messages generated by the MONITR process (2) being sent to a remote console *tty1* (1).

Event messages are displayed immediately even though you may be in the middle of typing a command. The display and keyboard have separate buffers to prevent command confusion. The following example shows the MONITR process connected to TTY1 after executing the **divert 2 1** command. If you want to stop the output, enter **halt 2**. The **halt** command is discussed at "Halt" on page 2-6.

```
Copyright Notices:
Copyright IBM Corp. 1994, 1997

MOS Operator Control

* divert 2 1

* status
  Pid  Name      Status TTY  Comments
  1    COpCon    IOW    TTY0 gzs
  2    Monitr    IDL    TTY0
  3    Tasker    RDY    --
  4    MOSDBG    DET    --
  5    CGWCon    DET    --
  6    Config    DET    --
  8    ROpCon    IDL    TTY1
  9    ROpCon    RDY    TTY2 jlg@128.185.40.40
  10   CES3      IDL    --
  11   TOUT      IDL    --
  12   L2S3      IDL    --
  13   L3L2      IDL    --
  14   LLL2      IDL    --
  15   S3CE      IDL    --
```

Syntax: <u>d</u>ivert   *pid tty#*

Example: `divert 2 1`

Flush

Use the **flush** command to clear the output buffers of the MONITR process. This command is generally used prior to displaying the contents of the MONITR's FIFO buffer to prevent messages from scrolling off the screen. Accumulated messages are discarded.

The router allows only certain processes to be redirected. To obtain the *pid* and *tty#*, use the OPCON **status** command. As you can see in the following example, after executing the **flush 2** command, the output of the MONITR process is sent to the SNK (it has been flushed).

Syntax: <u>f</u>lush   *pid*

Example: `flush 2`

```
    * flush 2

    * status
      Pid  Name     Status TTY  Comments
      1    COpCon   IOW    TTY0 gzs
      2    Monitr   IDL    SNK
      3    Tasker   RDY    --
      4    MOSDBG   DET    --
      5    CGWCon   DET    --
      6    Config   DET    --
      8    ROpCon   IDL    TTY1
      9    ROpCon   RDY    TTY2 jlg@128.185.40.40
```

Halt

Use the **halt** command to suspend all subsequent output from a specified process until the **divert**, **flush**, or **talk** OPCON command is issued to the process. The router cannot redirect all processes. **Halt** is the default state for output from a process. To obtain the PID for this command, use the OPCON **status** command. As you can see in the following example, after executing the **halt 2** command, the MONITR process is no longer connected to TTY1. Event messages no longer appear.

Syntax: halt *pid*

Example: `halt 2`

```
    * halt 2

    * status
    Pid  Name     Status TTY  Comments
    1    COpCon   IOW    TTY0 gzs
    2    Monitr   IDL    --
    3    Tasker   RDY    --
    4    MOSDBG   DET    --
    5    CGWCon   DET    --
    6    Config   DET    --
    8    ROpCon   IDL    TTY1
    9    ROpCon   RDY    TTY2 jlg@128.185.40.40
```

Intercept

Use the **intercept** command to change the OPCON intercept character. The intercept character is what you enter from other processes to get back to the OPCON process. The default intercept key combination is `Ctrl` `P`.

The intercept character *must* be a control character. Enter the ^ (shift 6) character followed by the letter character you want for the intercept character.

Note: Do not set the intercept character to the return key or to a printable character.

Syntax: intercept   *character*

Example: `intercept ^u`

From this example, you will have an intercept character of `Ctrl` `U`.

Logout

Use the **logout** command to terminate the current session for the user who enters the logout command. If the console login is enabled, this command will require the next user to log in using an authorized userid/password combination. If the console login is not enabled, the OPCON prompt appears again.

Syntax: <u>logo</u>ut

Example: `logout`

Memory

Use the **memory** command to obtain and display information about the router's global heap memory usage. The display helps you to determine if the router is being utilized efficiently. For an example of memory utilization, see Figure 2-3.

Syntax: <u>m</u>emory

Example: `memory`

`Number of bytes:  Busy = 319544, Idle = 1936, Free = 1592`

Busy Specifies the number of bytes currently allocated.

Idle Specifies the number of bytes previously allocated but freed and available for reuse.

Free Specifies the number of bytes that were never allocated from the initial free storage area.

Note: The sum of the Idle and Free memory equals the total available heap memory.

```
 _____          _____          _____
|                |        |                |        | Free=1128      |
| Free=1393515   |        | Free=103068    |        |----------------|
|                |        |                |        |                |
|                |        |----------------|        | Idle=103068    |
|----------------|←Water  | Idle=1128      |        |                |
| Idle=1128      | Mark   |                |        |_____|
|_____|        |_____|        |                |
|                |        |                |        |                |
| Busy=103068    |        | Busy=1393515   |        | Busy=1393515   |
|_____|        |_____|        |_____|

   Under-Utilized            Properly Utilized          Over-Utilized
```
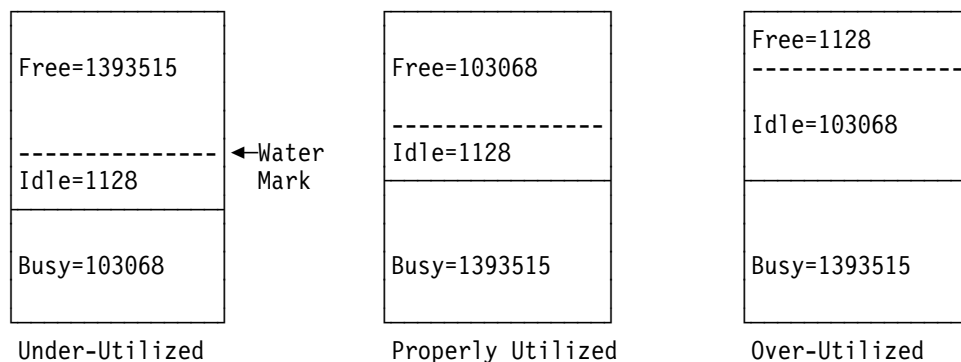
Figure 2-3. Memory Utilization

Reload

Use the **reload** command to reboot the router by loading in a new copy of the router software. When you use this command from a remote console, you install a new software load without going to the router. This command executes the same functions as pressing the reset button except that the router will not dump (if so configured). Before the reload takes effect, you are prompted to confirm the reload.

Syntax: <u>rel</u>oad

Example: `reload`

```
Are you sure you want to reload the gateway (Yes or No)?
```

Status

Use the **status** command to display information about all router processes. By entering the PID after the **status** command, you can select to look at the status of only the desired process. The following example shows the total status display.

Syntax: s̲tatus   *pid*

Example: status

```
Pid  Name      Status TTY  Comments
1    COpCon    IOW    TTY0
2    Monitr    IDL    --
3    Tasker    RDY    --
4    MOSDBG    DET    --
5    CGWCon    IOW    --
6    Config    IOW    TTY1
8    ROpCon    IOW    TTY1 128.185.46.101
9    ROpCon    RDY    TTY2 128.185.46.104
```

Pid   Specifies the PID. This is the process to talk to or from OPCON, or it can be an argument to the STATUS command to request status information about a specific process.

Name   Specifies the process name. It usually corresponds to the name of the program that is running in the process.

Status   Specifies one of the following:

IDL   Specifies that the process is idle and waiting for completion of some external event, such as asynchronous I/O.

RDY   Specifies that the process is ready to run and is waiting to use the CPU.

IOW   Specifies that the process is waiting for synchronous I/O, usually its expected standard input, to complete.

DET   Specifies that the process has output ready to be displayed and it is either waiting to be attached to a display console or waiting to have its output diverted to a specified console.

FZN   Specifies that the process is frozen due to an error. This usually means the process is trying to use a device which is faulty or incorrectly configured.

TTYn   Specifies the output terminal, if any, to which the process is currently connected.

TTY0   Local console

TTY1 or TTY2   Telnet consoles.

SNK   Process has been flushed.

Two dashes (--)   Process has been halted.

Comments

Specifies the user's login IP address provided during login when a user is logged in using Telnet (ROpCon).

Talk

Use the **talk** command to connect to other router processes, such as GWCON, MONITR, or CONFIG. After connecting to a new process, you can send specific commands to and receive output from that process. You cannot talk to the TASKER or OPCON process.

To obtain the PID, use the OPCON **status** command. Once you are connected to the second-level process, such as CONFIG, use the intercept character, **Ctrl** **P**, to return to the * prompt.

Syntax: t̲alk   *pid*

Example: `talk 5`

When using third-level processes, such as IP Config or IP, use the **exit** command to return to the second level.

Telnet

Use the **telnet** command when you want to remotely attach to another router or to a remote host (*ip address*). The only optional parameter is the terminal type that you want to emulate.

A router has a maximum of five Telnet sessions: two servers (inbound to the router), and three clients (outbound from the router).

Note: To use Telnet in a pure bridging environment, you must enable Host Services.

Syntax: t̲e̲lnet   *ip address terminal type*

Example: `telnet 128.185.10.30`
                or
        `telnet 128.185.10.30  23`
                or
        `telnet 128.185.10.30  vt100`

```
Trying 128.185.10.30 ...
Connected to 128.185.10.30
Escape character is '^]'
```

If you are Telneting to a non-existent IP address, the router displays:

```
Trying 128.185.10.30 ...
```

To enter the Telnet command mode, type the escape character-sequence, which is **Ctrl-]**, at any prompt.

```
telnet>
```

If you telnet into a router,

- Press ← **Backspace** to delete the last character typed on the command line.

  Note: If you use a VT100t terminal, do not press ← **Backspace** because it inserts invisible characters. You must press **Delete** to delete the last character.

- Press **Ctrl** **U** at the `telnet>` prompt to delete the whole command line entry so that you can re-enter a command.

The Telnet command mode consists of the following subcommands:

close   Close current connection
display Display operating parameters
mode   Try to enter line-by-line or character-at-a-time mode
open   Connect to a site
quit   Exit Telnet
send   Transmit special characters ('send ?' for more)
set   Set operating parameters ('set ?' for more)
status   Print status information
toggle   Toggle operating parameters ('toggle ?' for more)
z   Suspend Telnet
?   Print help information

The **status** and **send** subcommands have one of two responses depending on whether or not the user is connected to another host. For example:

Connected to a host:

```
telnet>   status

Connected to 128.185.10.30
Operating in character-at-a-time mode.
Escape character is ^].

telnet>   send ayt
```

Note: The send command currently supports only ayt.

Not connected to a host:

```
telnet>   status

Need to be connected first.

telnet>   send ayt

Need to be connected first.
```

Use the **close** subcommand to close a connection to a remote host and terminate the Telnet session. Use the **quit** subcommand to exit the **telnet** command mode, close a connection, and terminate a Telnet session.

```
telnet>   close

 or

telnet>   quit

logout
*
```

Chapter 3. The CONFIG Process and Commands

This chapter describes the CONFIG process and includes the following sections:

- "What is CONFIG?"
- "CONFIG-Only Mode" on page 3-2
- "Quick Configuration" on page 3-3
- "Configuring User Access" on page 3-5
- "Entering and Exiting CONFIG" on page 3-6
- "CONFIG Commands" on page 3-7

What is CONFIG?

The Configuration process (CONFIG) is a second-level process of the router user interface. Using CONFIG commands, you can:

- Set or change various configuration parameters
- Add or delete an interface to the hardware configuration
- Enter the Boot CONFIG command mode
- Enter the Quick Configuration mode
- Clear, list, or update configuration information
- Enable or disable console login and modem control
- Communicate with third-level processes, including protocol environments

CONFIG lets you display or change the configuration information stored in the router's non-volatile configuration memory. Changes to system and protocol parameters do not take effect until you reload the router software. (For more information, refer to the OPCON **reload** command in Chapter 2, "The OPCON Process and Commands").

Note: The changes you make through CONFIG are retained in a configuration database in non-volatile configuration memory. They are retained during power-downs and are recalled when you restart the router.

CONFIG fits into the router software structure as shown in Figure 3-1 on page 3-2.

Router Software Processes
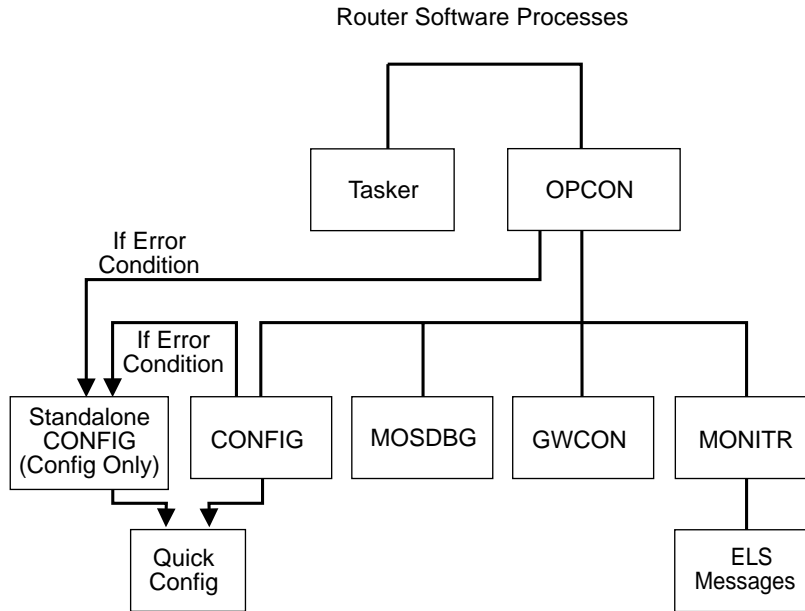
Figure   3-1. CONFIG in the Router Software Structure

The CONFIG command interface is made up of levels that are called modes. Each mode has its own prompt. For example, the prompt for the TCP/IP protocol is `IP config>`.

If you want to know the process and mode you are communicating with, press **Return** to display the prompt. Some commands in this chapter, such as the **network** and **protocol** commands, allow you to access and exit the various levels in CONFIG. See Table 3-2 on page 3-7 for a list of the commands you can issue from the CONFIG process.

CONFIG-Only Mode

CONFIG-Only mode is a way to back out of a bad configuration that is causing the router to crash during start-up. Use the CONFIG-Only mode *only* to change devices or data links (that is, for unsupported devices) or to reduce memory use (for *no memory* crashes) such as routing table sizes, packet sizes, and receive buffer allocations.

Note: CONFIG-only is provided only for getting a subset of configuration commands when a config problem causes the router to panic, check, fatal, or bughlt. Do *not* use CONFIG-Only mode for general router configuration; many of the device-related commands are disabled in CONFIG-ONLY mode and some may cause a crash.

Automatic Entry Into Config-Only Mode

CONFIG-Only mode is entered when the router detects a problem during operation or during router initialization.

Any of the following situations will cause the router to enter into CONFIG-Only mode:

- The software load does not match the device configuration. More particularly, an attempt is made to configure a device or data link that is unsupported by the software load.

- Devices are configured but there are no protocols configured.

- Deletion of all router interface information.

If the router entered into the configuration-only mode because an unsupported device has been configured:

- Change the device information to match the hardware installed in (and supported by) the router.
- Enter the **Reload** command from the `Config (only)>` prompt.
- The router will automatically enter into OPCON (*).

Manual Entry Into Config-Only Mode

To enter the config-only mode:

- Reload the router with no configuration.
- Reload the router with no interfaces configured.
- Reload the router with no protocols configured.

Any of the above will force the box into config-only.

Otherwise, the router will reboot. If boot information is missing, the software will load from the hard drive.

See Chapter 5, "Accessing the Firmware from the Command Line Interface" for more detail.

During initial start-up, if no devices are configured the router comes up in CONFIG-ONLY mode. If no protocols are configured the router comes up in CONFIG-ONLY mode and automatically enters Quick Config. Quick Configuration is explained in the next section.

Quick Configuration

Quick Configuration (Quick Config) provides a minimal set of commands that allow you to configure bridging protocols and routing protocols present in the router load.

Attention At least one network device must be configured before using quick config. To add a configuration use the **add device** command at the `config(only)>` or `config>` prompt.

Table 3-1 lists what Quick Config supports.

Table 3-1. Quick Config Capabilities	
Bridging protocols	**Routing Protocols**
ATM	STB, SRT, SRB, SR/TB
IP, IPX, DNA IV	

The Quick Config complements the existing configuration process by offering a shortcut. This shortcut allows you to configure the minimum number of parameters for these bridging protocols and routing protocols without having to exit and enter the different configuration processes. The other parameters are set to selected defaults.

Situations that call for the router to be quickly configured are:

- Blank or corrupted configuration memory, such as when one of the following situations occurs:
  - The router is configured for the first time.
  - Voltage fluctuations resulted in corruption of the hard file.

- Demonstration purposes, for which the router needs to be quickly configured to demonstrate its capabilities.

- Bench-marking tests to get the tests going without having to learn the router's operating system commands.

Quick Config operates as follows:

- It asks a series of questions with default values.

- It offers a short-cut to the detailed configuration of the normal mode command set.

Quick Config sets a number of default parameters based upon how you answer the configuration questions.

What cannot be configured with Quick Config can be configured using Config after exiting it.

You cannot delete Quick Config information from within Quick Config. However, you can correct information either by exiting and returning to Quick Config, or by entering the **reload** command as a response to some Quick Config questions.

For complete information on using the Quick Config software, refer to the *Quick Configuration* information provided in the *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*.

Manual Entry Into the Quick Config Mode

You might want to get to Quick Config manually to demonstrate the router's capabilities, reconfiguring on the fly to benchmark tests without having to learn the router's operating system commands.

To enter Quick Config, type **qconfig** at the `Config>` prompt.

Exiting from Quick Config Mode

To exit Quick Config, restart by entering **r** from any prompt. Follow the queries until you enter **no** and then enter **q** to quit. The router returns to either the `Config (only)>` or the `Config>` prompt.

```
***********************************************************
Bridging Configuration
***********************************************************

Type 'Yes' to Configure Bridging
Type 'No' to skip Bridging Configuration
Type 'Quit' to exit Quick Config

Configure Bridging? (Yes, No, Quit): [Yes] y
Type 'r' any time at this level to restart Bridging Configuration

STB will be enabled on all LAN interfaces

Configure SRT Bridging? (Yes, No): [Yes] r

Restarting Bridging Configuration

Configure Bridging? (Yes, No): [Yes] n

***********************************************************
Protocol Configuration
***********************************************************

Type 'Yes' to Configure Protocols
Type 'No' to skip Protocol Configuration
Type 'Quit' to exit Quick Config

Configure Protocols? (Yes, No, Quit): [Yes] q

Quick Config Done

Config>
```

Configuring User Access

The router configuration process allows for a maximum of 50 user names, passwords, and levels of permission. Each user added needs to be assigned a password and level of permission. There are three levels of permission: *Administration, Operation,* and *Monitoring*.

For more information, see the **add user** command on page 3-10.

Technical Support Access

If you are the system administrator, when you add a new user for the first time, you are asked if you want to add Technical Support access. If you answer yes, Technical Support access is granted the same privileges as Administrator access.

The password for this account is automatically selected by the software and is known by your service representative. This password can be changed using the **change user** command; however, if you do change the password, customer service cannot provide remote support. For additional information on the use of the **change user** command, see page 3-18.

Entering and Exiting CONFIG

To enter CONFIG from OPCON (*):

1. At the OPCON prompt, enter the **status** command to find the PID of CONFIG. (See page 1-5 for a sample output of the **status** command.)

   `* status`

2. Enter the OPCON **talk** command and the PID for CONFIG:

   `* talk 6`

The console displays the CONFIG prompt (`Config>`). Now, you can enter CONFIG commands. If the prompt does not appear, press the **Return** key again. To exit CONFIG and return to the OPCON prompt (*), enter the intercept character. (The default is **Ctrl** **P** .)

Entering the Desired Protocol Configuration Process

For information on accessing a particular protocol's configuration process, see "Protocol" on page 3-28.

CONFIG Commands

This section describes each of the CONFIG commands. Each command includes a description, syntax requirements, and an example. The CONFIG commands are summarized in Table 3-2.

After accessing the CONFIG environment, enter the configuration commands at the Config> prompt.

Table 3-2. CONFIG Command Summary

Command	Function
? (Help)	Lists the CONFIG commands or lists the options associated with specific commands.
Add	Adds an interface to the router configuration, or a user to the router.
Boot	Enters Boot CONFIG command mode.
Change	Changes a user's password or a user's parameter values associated with this interface. Also changes a slot/port of an interface.
Clear	Clears configuration information.
Delete	Deletes an interface from the router configuration or deletes a configured user.
Disable	Disables login from a remote console, or a specified interface.
Enable	Enables login from a remote console, enables modem use, or enables a specified interface.
Event	Enters the Event Logging System configuration environment.
Feature	Provides access to configuration commands for independent router features outside the usual protocol and network interface configuration processes.
List	Displays system parameters, hardware configuration, a complete user list.
Network	Enters the configuration environment of the specified network.
Patch	Modifies the router's global configuration.
Protocol	Enters the command environment of the specified protocol.
Qconfig	Initiates the Quick Config process.
Set	Sets system-wide parameters for buffers, host name, inactivity timer, packet size, prompt level, location, and contact-person.
Time	Keeps track of system time and displays it on the console.
Unpatch	Restores patch variables to default values.
Update	Updates the current version of the configuration.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax:	?
Example:	**?**
	ADD
	BOOT
	CHANGE
	CLEAR
	DELETE
	DISABLE
	ENABLE
	EVENT
	FEATURE
	LIST
	NETWORK
	PATCH
	PROTOCOL
	QCONFIG
	SET
	TIME OF DAY PARAMS
	UNPATCH
	UPDATE
Example:	**list ?**
	devices
	configuration
	patches
	users
	v25-bis-address

Add

Use the **add** command to add an interface to the configuration, or user-access.
This command also recreates device records if the configuration is inadvertently
lost.

Syntax:	<u>a</u>dd	<u>d</u>evice . . .
		<u>i</u>sdn-address . . .
		<u>p</u>pp-user
		<u>u</u>ser . . .
		<u>v</u>25-bis-address

device   *device_type*   *additional-config-info*

> With the **add device** command, you must enter the interface device type
> (*device_type*). You are prompted for additional configuration parameters.
> This additional information varies by device and platform. Refer to "Accessing
> Network Interface Configuration and Console Processes" on page 1-15 for
> additional information about device type and configuration parameters.

> **Note:** If you are adding more than one interface, the order in which you add
> them is important because the router assigns a sequential interface number to
> the device when it is added. This interface number is an index number in the
> device list; it links the device with other protocol configuration information,
> such as the IP addresses associated with the device. (For more information,
> refer to the **list devices** command on page 3-23.)

> All device and protocol configuration information related to network interfaces
> is stored by interface number. Any changes made to interface numbers will
> invalidate much of the device configuration information in the protocols.

Example: `add device atm`

```
Device Slot #(1-8) [1]? 2
Adding CHARM ATM Adapter device in slot 2 port 1 as interface x
(where x is the interface number assigned)
```

To determine which devices you can add, use the **add devices ?** command.

IBM 2216 Device Support Restrictions: The following rules apply when adding devices to the IBM 2216:

- As many as two ATM interfaces may be defined.

- As many as four ISDN-PRI interfaces may be defined.

- Adding a LAN device in slot 3, 4, 7, or 8 will disable the other slot in the pair. For example, if you add a LAN device to slot 4, slot 3 will be disabled. Likewise, if you add the device to slot 3, slot 4 will be disabled. The same rules apply for adding a LAN device in slots 7 and 8.

- When adding interfaces with more than one port, add one port at a time. Use the **add device** command twice to add both ports of a dual-port adapter, for example.

- All interfaces on a V.35/V.36 adapter must use either V.35 cables or V.36 cables. The type of fanout cable attached to the V.35/V.36 adapter determines which type of cables (V.35 or V.36) can be used.

isdn-address *address-name   network-dial-address   network-subdial-address*
   Adds the local and remote numbers of the ISDN end-points that will be communicating with your router.

   address-name
      Can be anything (such as a description of the port).

   network-dial-address
      The telephone number of the local or the destination port.

   network-subdial-address
      The additional part of the telephone number, such as an extension, that gets interpreted when the interface connects to a PBX; this parameter is optional.

   Note: You can use punctuation, such as parentheses and dashes, but the punctuation is not significant (the router uses only the numbers).

   Example: `add isdn-address line 1 local`

   ```
   Assign network dial address [0 - 32 digits]? 1 2345 67
   Assign network subdial address [0 - 19 digits]? 98765
   ```

ppp_user
   Adds a username to the PPP user data base. You need to configure PPP users if you are using PPP authentication protocols. Refer to "PPP Authentication Protocols" on page 28-7 for more details. You add a PPP user for each remote router that can connect to the router you are configuring. The PPP user name (ID) and password are used during PPP authentication. You can add as many as 500 PPP users.

Example:　　　　**add ppp_user password haag**

```
Config> add ppp_user password haag
password (6 - 32 characters):
Enter password again:
user [haag] added successfully
```

user *user_name*

Gives a user access to the router. You can authorize up to 50 users to access the router. Each user_name is eight characters and is case-sensitive.

When the first user is added, console login is automatically enabled. Each user added must be assigned one of the following permission levels. Table 3-3 describes the different access privileges that are available.

Table 3-3. Access Privileges	
Level of Access	**Description**
Administrator (A)	Displays configuration and user information, adds/modifies/deletes configuration and user information. The Administrator can access any router function.
Operator (O)	Views router configuration, views statistics, runs potentially disruptive tests, dynamically changes router operation, and restarts the router. Operators cannot modify the permanent router configuration. All actions can be undone with a system restart.
Monitor (M)	Views router configuration and statistics but cannot modify or disrupt the operation of the router.
Tech Support	Allows your service representative to gain access to the router if a password is forgotten. Cannot be assigned to users.

Note: To add a user, you must have administrative permission. You do not have to reinitialize the router after adding a user.

Example:　　　　**add user John**

```
Enter password:
Enter password again:
Enter permission (A)dmin, (O)perations, (M)onitor [A]?
Do you want to add Technical Support access?
(Yes or [No]):
```

Enter password　　　　Specifies the access password for the user. Limited to 80 alphanumeric characters and is case-sensitive.

Enter password again　　　　Confirms the access password for the user.

Enter permission　　　　Specifies the access permission for the user: A, O, or M. (See Table 3-3.)

v25-bis-address
Adds the local and remote numbers of the V.25 bis end-points that will be communicating with the router. The network *address-name* can be anything, such as a description of the port. You can use any string of up to 23 printable ASCII characters. The *network-dial-address* is the telephone number of the local or destination port. For more information, see Chapter 35, "Using and Configuring the V.25 bis Network Interface" on page 35-1.

Note: You can use punctuation, such as parentheses and dashes, but the punctuation is not significant (the router uses only the numbers).

Example:　　　　`add v25-bis-address remote-site baltimore`
`1-909-555-0983`

Boot

Use the **boot** command to enter the Boot CONFIG command environment. For Boot CONFIG information, see Chapter 4, "Performing Change Management Operations" on page 4-1.

Syntax: <u>b</u>oot

Example:　　　　`boot`

```
TFTP Boot/dump configuration
Boot config>
```

Change

Use the **change** command to modify an interface in the configuration, change your own password, or change user information.

Syntax: <u>ch</u>ange　　　　　　<u>d</u>evice . . .
<u>pa</u>ssword
<u>pp</u>p_user . . .
<u>u</u>ser

device　*device_type*
With the **change device** command you can:

- Change the slot of an existing interface. (Change slot x in interface record n to y where slot y is unoccupied.)

- Change the port of an existing interface. (Change port x in interface record n to y where port y is unoccupied.)

- Swap slots of two existing interfaces. (Swap slot x and slot y in interface records with x or y.)

- Swap ports of two existing interfaces. (Swap port u and slot x in one interface record with port v and slot y in another interface record of the same hardware type.)

- Replace the slot in an existing interface with the slot in another. (Interface configuration for slot x will become interface configuration for slot y. Interface records for slot y will be deleted.)

- Replace the port of one existing interface with the port of another. (Interface configuration for slot x port u will become interface configuration for slot y port v. The interface record for slot y port v will be deleted.)

When the target slot is occupied:

1. If the "swap" option is selected, the source and target slots are swapped in all the interface records in which they appear.

2. If the "replace" option is selected, the interface configuration for slot x will become the interface configuration for slot y. Interface records for slot y will be deleted.

When the target port is occupied:

1. If the "swap" option is selected, the source and target ports can be swapped in their respective interface records if their hardware types in these interface records are identical. For example, 8 port EIA 232E/V.24.

2. If the "replace" option is selected, the interface configuration for slot x port u will become the interface configuration for slot y port v. The interface record for slot y port v will be deleted.

Note: An Ethernet or Token Ring adapter card requires two empty slots if it is installed in slot 3, 4, 7, or 8. Therefore a Token Ring or Ethernet adapter can be installed in slot 3 or 4 (or slot 7 or 8) only if both slots 3 and 4 (or 7 and 8) are unoccupied.

If you try to change the slot of a Token Ring or Ethernet adapter to slot 3 or 4 (or 7 or 8) when both 3 and 4 (or 7 and 8) are not unoccupied, the change is not accepted and a warning message is issued as shown in the example "Change slot 6 on interface 1 to unoccupied slot 8" on page 3-12.

Example: Change slot 5 on interface 0 to unoccupied slot 7.

```
Config>list dev
Ifc 0 Token Ring                     Slot: 5  Port: 1
Ifc 1 Token Ring                     Slot: 6  Port: 1
Ifc 2 Token Ring                     Slot: 1  Port: 1
Ifc 3 8 port EIA-232E/V.24 PPP       Slot: 2  Port: 0
Ifc 4 8 port EIA-232E/V.24 PPP       Slot: 2  Port: 1
Ifc 5 8 port EIA-232E/V.24 PPP       Slot: 2  Port: 2
Ifc 6 6 port V.35/V.36 PPP           Slot: 3  Port: 0
Ifc 7 6 port V.35/V.36 PPP           Slot: 3  Port: 5
Ifc 8 8 port EIA-232E/V.24 PPP       Slot: 4  Port: 0

Config>change device
Which configured slot would you like to change? (1, 2, 3, 4, 5, 6)[1]? 5
Change all ports on slot # 5 (Yes or No)? [Yes]: y
Which slot would you like to change to? (1-8)  [1]? 7

Changed slot 5 to slot 7 in 1 intf (port) record...

Config>list dev
Ifc 0 Token Ring                     Slot: 7  Port: 1
Ifc 1 Token Ring                     Slot: 6  Port: 1
Ifc 2 Token Ring                     Slot: 1  Port: 1
Ifc 3 8 port EIA-232E/V.24 PPP       Slot: 2  Port: 0
Ifc 4 8 port EIA-232E/V.24 PPP       Slot: 2  Port: 1
Ifc 5 8 port EIA-232E/V.24 PPP       Slot: 2  Port: 2
Ifc 6 6 port V.35/V.36 PPP           Slot: 3  Port: 0
Ifc 7 6 port V.35/V.36 PPP           Slot: 3  Port: 5
Ifc 8 8 port EIA-232E/V.24 PPP       Slot: 4  Port: 0
```

Interface 0 is now changed to slot 7.

Example: Change slot 6 on interface 1 to unoccupied slot 8.

```
Config>change device
Which configured slot would you like to change? (1, 2, 3, 4, 6, 7)[1]? 6
Change all ports on slot # 6 (Yes or No)? [Yes]: y
Which slot would you like to change to? (1-8)  [1]? 8

Cannot add Token Ring to slot 8.
Slot 7 is occupied so Token Ring cannot be added in slot 8.
```

Note: See page 3-12 for requirements for changing slots 3, 4, 7, and 8 for Token Ring or Ethernet.

Example: Swap slot 6 on interface 1 to occupied slot 1.

```
Config>change device
Which configured slot would you like to change? (1, 2, 3, 4, 6, 7) [1] 6
Change all ports on slot # 6 (Yes or No)? [Yes]: y
Which slot would you like to change to? (1-8)  [1]? 1

Configuration for slot 1 already exists. You can:
a - abort this operation
r - replace configuration
    (Interface configuration for slot 6 will become interface
     configuration for slot 1.  Interface records for slot 1
     will be deleted!)
s - swap configuration (slot 1 will be swapped with slot 6.)
s

Swapped slot 6 with slot 1 in 1 port record...

Config>list dev
Ifc 0 Token Ring                   Slot: 7  Port: 1
Ifc 1 Token Ring                   Slot: 1  Port: 1
Ifc 2 Token Ring                   Slot: 6  Port: 1
Ifc 3 8 port EIA-232E/V.24 PPP     Slot: 2  Port: 0
Ifc 4 8 port EIA-232E/V.24 PPP     Slot: 2  Port: 1
Ifc 5 8 port EIA-232E/V.24 PPP     Slot: 2  Port: 2
Ifc 6 6 port V.35/V.36 PPP         Slot: 3  Port: 0
Ifc 7 6 port V.35/V.36 PPP         Slot: 3  Port: 5
Ifc 8 8 port EIA-232E/V.24 PPP     Slot: 4  Port: 0
```

The configurations for interface 2 and interface 1 are swapped.

Example: Change port 1 slot 5 to port 0 in unoccupied slot 2.

```
Config>change device
Which configured slot would you like to change? (1, 3, 4, 5, 7)  [1]? 5
Change all ports on slot # 5 (Yes or No)? [Yes]: n
Which port would you like to change in slot 5? (0, 1, 2) [0]? 1
Which slot would you like to change to? (1-8)  [1]? 2
Which port would you like port 1 in slot 5 to move to in slot2?#(0-7)[0]? 0

Changed slot 5 port 1 to slot 2 port 0...

Config>list dev
Ifc 0 Token Ring                      Slot: 7  Port: 1
Ifc 1 Token Ring                      Slot: 1  Port: 1
Ifc 2 8 port EIA-232E/V.24 PPP        Slot: 5  Port: 0
Ifc 3 8 port EIA-232E/V.24 PPP        Slot: 2  Port: 0
Ifc 4 8 port EIA-232E/V.24 PPP        Slot: 5  Port: 2
Ifc 5 6 port V.35/V.36 PPP            Slot: 3  Port: 0
Ifc 6 6 port V.35/V.36 PPP            Slot: 3  Port: 5
Ifc 7 8 port EIA-232E/V.24 PPP        Slot: 4  Port: 0
```

Interface 3, which was at slot 5 - port 1, is changed to slot 2 port 0.

Example: Change port 0 slot 2 to port 5 in slot 3 (different hardware types).

```
Config>change device
Which configured slot would you like to change? (1, 2, 3, 4, 5, 7) [1]? 2
Change all ports on slot # 2 (Yes or No)? [Yes]: n
Which port would you like to change in slot 2? (0) [0]? 0
Which slot would you like to change to? (1-8)  [1]? 3
Which port would you like port 0 in slot 2 to move to in slot 3? #(0-7) [0] 5

Aborting - source and target slots of different type.
```

Example: Change port 0 slot 2 to port 5 in slot 4 (same hardware types).

```
Config>change device
Which configured slot would you like to change? (1, 2, 3, 4, 5, 7) [1]? 2
Change all ports on slot # 2 (Yes or No)? [Yes]: n
Which port would you like to change in slot 2? (0) [0]? 0
Which slot would you like to change to? (1-8)  [1]? 4
Which port would you like port 0 in slot 2 to move to in slot 4? #(0-7)[0] 5

Changed slot 2 port 0 to slot 4 port 5...

Config>list dev
Ifc 0 Token Ring                      Slot: 7  Port: 1
Ifc 1 Token Ring                      Slot: 1  Port: 1
Ifc 2 8 port EIA-232E/V.24 PPP        Slot: 5  Port: 0
Ifc 3 8 port EIA-232E/V.24 PPP        Slot: 4  Port: 5
Ifc 4 8 port EIA-232E/V.24 PPP        Slot: 5  Port: 2
Ifc 5 6 port V.35/V.36 PPP            Slot: 3  Port: 0
Ifc 6 6 port V.35/V.36 PPP            Slot: 3  Port: 5
Ifc 7 8 port EIA-232E/V.24 PPP        Slot: 4  Port: 0
```

Interface 3 which was at port 0, slot 2 is changed to port 5, slot 4.

Example: Change port 5 slot 4 to port 1 in slot 4 (same slots).

```
Config>change device
Which configured slot would you like to change? (1, 3, 4, 5, 7) [1]? 4
Change all ports on slot # 4 (Yes or No)? [Yes]: n
Which port would you like to change in slot 4? (0, 5) [0]? 5
Which slot would you like to change to? (1-8)  [1]? 4
Which port would you like port 5 in slot 4 to move to in slot 4? #(0-7) [0] 1

Changed slot 4 port 5 to slot 4 port 1...

Config>list dev
Ifc 0 Token Ring                     Slot: 7  Port: 1
Ifc 1 Token Ring                     Slot: 1  Port: 1
Ifc 2 8 port EIA-232E/V.24 PPP       Slot: 5  Port: 0
Ifc 3 8 port EIA-232E/V.24 PPP       Slot: 4  Port: 1
Ifc 4 8 port EIA-232E/V.24 PPP       Slot: 5  Port: 2
Ifc 5 6 port V.35/V.36 PPP           Slot: 3  Port: 0
Ifc 6 6 port V.35/V.36 PPP           Slot: 3  Port: 5
Ifc 7 8 port EIA-232E/V.24 PPP       Slot: 4  Port: 0
```

Interface 3 is now at port 1, slot 4.

Example: Change (swap) port 1 slot 4 to occupied port 0 in slot 5.

```
Config>change device
Which configured slot would you like to change? (1, 3, 4, 5, 7) [1]? 4
Change all ports on slot # 4 (Yes or No)? [Yes]: n
Which port would you like to change in slot 4? (0, 1) [0]? 1
Which slot would you like to change to? (1-8)  [1]? 5
Which port would you like port 1 in slot 4 to move to in slot 5? #(0-7) [0] 0

Configuration for slot 5 (port 0) already exists. You can:
a - abort this operation
r - replace configuration
    (Interface record for slot 4 port 1 will become interface
     configuration for slot 5 port 0.  The interface record for
     slot 5 port 0 will be deleted!)
s - swap configuration (slot 5 port 0 will be swapped with slot 4
    port 1.)
s

Swapped slot 4 port 1 with slot 5 port 0...

Config>list dev
Ifc 0 Token Ring                     Slot: 7  Port: 1
Ifc 1 Token Ring                     Slot: 1  Port: 1
Ifc 2 8 port EIA-232E/V.24 PPP       Slot: 4  Port: 1
Ifc 3 8 port EIA-232E/V.24 PPP       Slot: 5  Port: 0
Ifc 4 8 port EIA-232E/V.24 PPP       Slot: 5  Port: 2
Ifc 5 6 port V.35/V.36 PPP           Slot: 3  Port: 0
Ifc 6 6 port V.35/V.36 PPP           Slot: 3  Port: 5
Ifc 7 8 port EIA-232E/V.24 PPP       Slot: 4  Port: 0
```

Interface 2 and interface 3 have swapped their port and slot configurations.

Example: Change (replace) port 1 slot 4 to occupied port 0 in slot 5.

```
Config>change device
Which configured slot would you like to change? (1, 3, 4, 5, 7) [1]? 4
Change all ports on slot # 4 (Yes or No)? [Yes]: n
Which port would you like to change in slot 4? (0, 1) [0]? 1
Which slot would you like to change to? (1-8)  [1]? 5
Which port would you like port 1 in slot 4 to move to in slot 5? #(0-7) [0] 0

Configuration for slot 5 (port 0) already exists. You can:
a - abort this operation
r - replace configuration
    (Interface configuration for slot 4 port 1 will become interface
    configuration for slot 5 port 0.  The interface record for
    slot 5 port 0 will be deleted!)
s - swap configuration (slot 5 port 0 will be swapped with slot 4
    port 1.)
r

Moved slot 4 port 1 to slot 5 port 0...

Config>list dev
Ifc 0 Token Ring                   Slot: 7  Port: 1
Ifc 1 Token Ring                   Slot: 1  Port: 1
Ifc 2 8 port EIA-232E/V.24 PPP     Slot: 5  Port: 0
Ifc 3 8 port EIA-232E/V.24 PPP     Slot: 5  Port: 2
Ifc 4 6 port V.35/V.36 PPP         Slot: 3  Port: 0
Ifc 5 6 port V.35/V.36 PPP         Slot: 3  Port: 5
Ifc 6 8 port EIA-232E/V.24 PPP     Slot: 4  Port: 0
```

Interface 2 is configured to slot 5 - port 0. The original interface 3 is deleted and the other interfaces are renumbered.

password

Modifies the password of the user who is now logged in.

Note: To change a user password, you must have administrative permission.

Example: **change password**

```
Enter current password:
Enter new password:
Enter new password again:
```

Enter current password   Specifies your current password.

Enter new password   Specifies your new password.

Enter new password again   Specifies your new password again for confirmation. If your confirmation does not match the previous new password, the old password remains in effect.

ppp_user

Changes the information for a specific PPP user.

Syntax: change ppp-user password

Example: **change ppp_user password haag**

```
Config> change ppp_user password haag
Enter password:
Enter password again:
user [haag] password updated successfully
```

user

Modifies the user information that was previously configured with the **add user** command.

Note: To change a user, you must have administrative permission.

Example: `change user`

```
User name: []
Change password? (Yes or No)
Change permission? (Yes or [No])
```

Clear

Use the **clear** command to delete the router's configuration information from non-volatile configuration memory.

Attention: Use this command only after calling your service representative.

To clear a process from non-volatile configuration memory, enter the **clear** command and the process name. To clear all information from configuration memory, except for device information, use the **clear all** command. If you want to clear all information, including the device information, use the **clear all** command and then the **clear device** command.

The **clear user** command clears all user information except the router console login information. This is left as enabled (if it was configured as enabled) even though the default value is "disabled."

Notes:

1. To clear user information, you must have administrative permission.

2. There may be other items in the list, depending upon what is included in the software load.

Syntax: <u>c</u>lear

<u>all</u>
<u>ap2</u> (AppleTalk 2)
<u>arp</u>   (ARP)
<u>asr</u>t   (Adaptive Source Route Protocol)
<u>app</u>n   (Advanced Peer-to-Peer Networking)
<u>atm</u>   (Asynchronous Transfer Mode)
<u>bgp</u>   (Border Gateway Protocol)
<u>b</u>oot
<u>dls</u> (Data Link Switching)
<u>dev</u>ice
<u>dial</u>er-circuit
<u>dn</u>   (DECnet)
<u>dvmr</u>p (Distance Vector Multicast Routing Protocol)
<u>e</u>ls (Event Logging System Information)
<u>f</u>r (Frame Relay)
<u>ip</u> (IP)
<u>ipx</u> (Novell IPX)
<u>is</u>dn
<u>os</u>i (OSI)
<u>ospf</u> (OSPF routing protocol)
<u>p</u>pp (Point-to-Point)
sdlc
<u>sl</u>c (Serial Line Controller SRAM block)
<u>sn</u>mp
<u>srb</u> (Source Route Bridge)
<u>srl</u>y (SDLC Relay)
<u>stb</u> (Spanning Tree Bridge)
<u>tcp</u>/ip-host
<u>tim</u>e (Time of day information)
<u>us</u>er
v25bis
<u>v</u>ines (Banyan VINES)
<u>w</u>rs (WAN Restoral feature)
<u>x2</u>5

Example: `clear ELS`

```
You are about to clear all Event Logging configuration information
Are you sure you want to do this (Yes or No):
```

Note: The previous message appears for any parameter configuration you are deleting.

Delete

Use the **delete** command to remove an interface from the list of devices stored in the configuration, or to remove a user. To use the **delete** command, you must have administrative permission.

Syntax: <u>de</u>lete

<u>i</u>nterface . . .
<u>is</u>dn-address
<u>p</u>pp_user . . .
<u>u</u>ser . . .
<u>v</u>25-bis-address

interface *intfc#*
>   To delete an interface, enter the interface or network number as part of the command. (Dial circuit is the only device type that can be deleted.) To obtain the interface number that the router assigns, use the **list device** command.

The delete interface command deletes the device configuration and any protocol information for that interface. However, the router will continue to run the previous configuration until it is reloaded.

When deleting a base ISDN interface or a base ATM interface all virtual interfaces running on that base net will also be deleted. So, any dial circuits configured on a base ISDN interface will be removed when the ISDN interface is deleted. Also, when deleting an ATM base net, all LAN Emulation Clients running on the base ATM interface will be deleted.

Example: `delete interface 2`

`Interface 2 was deleted successfully`

isdn-address *address-name*
> Removes a previously added ISDN address.

Example: `delete isdn-address remote-site-XYZ`

Note: If the *address-name* contains spaces (for example, **remote site XYZ**), you cannot enter the command on one line. Type `delete isdn-address` and press **Return**. Then enter the name when prompted.

ppp_user
> Deletes a user from the PPP user data base.

Example: `delete ppp_user haag`

```
Config> delete ppp_user haag
user [haag] deleted successfully
Config>
```

user *user_name*
> Removes user access to the router for the specified user.

Example: `delete user mary`

`Are you sure you want to delete user 'mary' ?(Y / N)_`

v25-bis-address *address-name*
> Removes a previously added V.25 bis address.

Example: `delete v25-bis-address remote-site-baltimore`

Note: If the *address-name* contains spaces (for example, **remote site Baltimore**), you cannot enter the command on one line. Type `delete v25-bis-address` and press **Return**. Then enter the name when prompted.

Disable

Use the **disable** command to prevent being prompted for a login from a remote console . The **disable** command also disables a specified interface. If the router has two service ports specify either **service1** or **service2**.

Syntax: <u>di</u>sable   <u>c</u>onsole-login
                                   <u>i</u>nterface . . .
                                   <u>d</u>ump-memory . . .
                                   <u>r</u>eboot-system . . .

console-login
> Disables the user from being prompted for a user ID and password on the
> physical console. The default is disabled.

> **Example:**   `disable console-login`

interface *interface#*
> Causes the specified interface to be disabled after issuing the **reload**
> command. The default is enabled.

> **Example:**   `disable interface 2`

dump-memory
> Disables the dumping of system memory to the installed hard disk when a
> serious error occurs.

> **Example:**   `Config> disable dump`
> `Disable system memory dumping? ([No]) Yes`

> System memory dump function disabled successfully.

reboot-system
> Disables the rebooting of the system when a serious error occurs. This may
> be desirable if the network service personnel wish to troubleshoot the error
> on-line. System rebooting cannot be disabled unless memory dumping is also
> disabled. If you attempt to disable system rebooting while memory dumping
> is enabled, system rebooting is aborted and the following message is
> displayed:

> `System reboot not disabled:  memory dumping must be disabled first`

> **Example:**   `Config> disable reboot`
> `Disable system reboot for serious errors? ([No]) Yes`

> System reboot function disabled successfully.

Enable

Use the **enable** command to allow login from a remote console, and enable a
specified interface.

Syntax: <u>en</u>able   <u>c</u>onsole-login
                                     <u>i</u>nterface . . .
                                     <u>d</u>ump-memory . . .
                                     <u>r</u>eboot-system . . .

console-login
> Enables the user to be prompted for a user ID and password on the physical
> console. This is useful for security situations. If you do not configure any
> administrative users and you enable this feature, the following message
> appears:

> `Warning: Console login is disabled until an`
> `administrative user is added.`

> By disabling the console login, a lock-out situation is prevented.

> **Example:**   `enable console-login`

interface *interface#*
>   Causes the interface to be enabled after issuing the **reload** command.

dump-memory
>   Enables the dumping of system memory to the installed hard disk when a
>   serious error occurs. This may be desirable so that the state of the unit at the
>   time of the error can be preserved for troubleshooting later. The dump
>   memory function cannot be enabled unless system rebooting is enabled. If
>   you attempt to enable the dump memory function while system rebooting is
>   disabled, dump memory is aborted and the following message is displayed:

```
System memory dump function not enabled:  rebooting must be enabled first
```

>   **Example:**   **Config> enable dump**
> ```
> Enable system memory dumping? ([No]) Yes
> ```
>   System memory dump function enabled successfully.

reboot-system
>   Enables the rebooting of the system when a serious error occurs.

>   **Example:**   **enable reboot-system**
> ```
> Enable system rebooting for serious errors? ([No]) Yes
> ```
>   System reboot function enabled successfully.

Event

Use the **event** command to enter the Event Logging System (ELS) environment so
that you can define the messages that will appear on the console. Refer to
Chapter 8, "Using and Configuring the Event Logging System (ELS)" for
information about ELS.

Syntax: event

Example: **event**
```
Event Logging System user configuration
ELS config>
```

Feature

Use the **feature** command to access configuration commands for specific router
features outside of the protocol and network interface configuration processes.

All 2216 features have commands that are executed by:

- Accessing the configuration process to initially configure and enable the feature, as well as perform later configuration changes.

- Accessing the console process to monitor information about each feature, or make temporary configuration changes.

The procedure for accessing these processes is the same for all features. The following information describes the procedure.

Enter a question mark after the **feature** command to obtain a listing of the features available for your software release.

Example: `feature ?`

```
WRS
BRS
MCF

Feature name or number [1] ?
```

To access a feature's configuration prompt, enter the **feature** command followed by the feature number or short name. Table 3-4 lists available feature numbers and names.

Table 3-4. IBM 2216 Feature Numbers and Names		
Feature Number	**Feature Short Name**	**Accesses the following feature configuration process**
0	WRS	WAN Restoral/Reroute
1	BRS	Bandwidth Reservation
2	MCF	MAC Filtering

Example: `feature mcf`

```
MAC filtering user configuration
Filter Config>
```

Once you access the configuration prompt for a feature, you can begin entering specific configuration commands for the feature. To return to the CONFIG prompt, enter the **exit** command at the feature's configuration prompt.

Syntax: f̲eature   *feature# OR feature-short-name*

Example: `feature 2`
                or
          `feature MCF`

List

Use the **list** command to display configuration information for all network interfaces, or configuration information for the router.

Syntax: l̲ist   d̲evices
                              c̲onfiguration
                              i̲sdn-address
                              p̲atches
                              ppp_users
                              u̲sers
                              v̲25-bis-address

devices
> Displays the relationship between an interface number and the hardware interface. You can also use this command to check that a device was added correctly issuing the **add** command.

> **Example:**

> **list devices**

```
Ifc 0 Token Ring                      Slot: 1  Port: 1
Ifc 1 Token Ring                      Slot: 1  Port: 2
Ifc 2 Token Ring                      Slot: 2  Port: 1
Ifc 3 Token Ring                      Slot: 2  Port: 2
Ifc 4 Ethernet                        Slot: 4  Port: 1
Ifc 5 Ethernet                        Slot: 4  Port: 2
Ifc 6 Ethernet                        Slot: 5  Port: 1
Ifc 7 Ethernet                        Slot: 5  Port: 2
Ifc 8 Ethernet                        Slot: 6  Port: 1
Ifc 9 Ethernet                        Slot: 6  Port: 2
Ifc 10 V.35/V.36 Frame Relay          Slot: 8  Port: 0
Ifc 11 V.35/V.36 X.25                 Slot: 8  Port: 1
Ifc 12 V.35/V.36 PPP                  Slot: 8  Port: 2
Ifc 13 V.35/V.36 PPP                  Slot: 8  Port: 3
Ifc 14 V.35/V.36 PPP                  Slot: 8  Port: 4
Ifc 15 V.35/V.36 PPP                  Slot: 8  Port: 5
```

configuration
> Displays configuration information about the router.

> **Example:**

> **list configuration**

```
Config>list conf
Hostname: -none-
Maximum packet size: -autoconfigured-
Maximum number of global buffers: -autoconfigured-
Console inactivity timer (minutes): 0
Physical console login: disabled
System rebooting on error: disabled
System memory dumping: disabled
Contact person for this node: -none-
Location of this node: -none-

Configurable Protocols:
Num Name  Protocol
0   IP    DOD-IP
3   ARP   Address Resolution
4   DN    DNA Phase IV
6   VIN   Banyan Vines
7   IPX   NetWare IPX
8   OSI   ISO CLNP/ESIS/ISIS
9   DVM   Distance Vector Multicast Routing Protocol
10  BGP   Border Gateway Protocol
11  SNMP  Simple Network Management Protocol
12  OSPF  Open SPF-Based Routing Protocol
20  SDLC  SDLC/HDLC-Relay
22  AP2   AppleTalk Phase 2
23  ASRT  Adaptive Source Routing Transparent Enhanced Bridge
24  HST   TCP/IP Host Services
25  LNM   LAN Network Manager
26  DLS   Data Link Switching
27  XTP   X.25 Transport Protocol

Configurable Features:
Num Name  Feature
0   WRS   WAN Restoral
1   BRS   Bandwidth Reservation
2   MCF   MAC Filtering
```

```
129600 bytes of configuration memory free
```

isdn-address
>   Displays the current ISDN address configurations.

Example: `list isdn-address`

```
Address assigned name    Network Address    Network Subdial Address
--------------------     --------------     -----------------------
remote site XYZ           1 2345 67          98765
```

patches
>   Displays the values of patch variables that have been entered using the
>   **patch** command.

Example: `list patches`

```
Patched variable        Value

ping-size               60
ping-ttl                59
ip-default-ttl          60
ethernet-security       3
rip-static-suppress     3
```

ppp_users
>   Lists the complete PPP user data base.

Example: `list ppp_users`

```
Config> list ppp_users

PPP User Name
------------
haag
dillon
```

users
>   Displays the users configured to access the system.

Example: `list users`

```
USER         PERMISSION
joe          operations
mary         administrative
peter        monitor
```

v25-bis-address
>   Displays the current V.25 bis address configurations. The V.25 bis address
>   configuration consists of the network address and network address name for
>   a local port (serial line interface) or destination port. The network address is
>   the telephone number of the local or destination port. The network address
>   name can be anything, such as the description of the port. For more
>   information, see Chapter 35, "Using and Configuring the V.25 bis Network
>   Interface" on page 35-1.

Example: `list v25-bis-address`

```
Address assigned name           Network Address
--------------------            --------------
v25-1                           8982800
v25-2                           8980001
westboro                        1-666-555-4444
```

Network

Use the **network** command to enter the network interface configuration environment for supported networks. Enter the interface or network number as part of the command. (To obtain the interface number, use the CONFIG **list device** command.) The appropriate configuration prompt (for example, TKR Config>) will be displayed. See the network interface configuration chapters in this book for complete information on configuring your types of network interfaces.

Note: Whenever you change a user-configurable parameter, you must **reload** the router for the change to take effect. To do so, enter the **reload** command at the OPCON prompt (*).

Syntax: network *interface#*

Example: `network 0`
                    `TKR config>`

Note: Not all network interfaces are user-configurable. For interfaces that you cannot configure, you receive the message: `That network is not configurable.`

Patch

Use the **patch** command for modifying the router's global configuration. Patch variables are recorded in nonvolatile configuration memory and take effect immediately; you do not have to wait for the next restart of the router. This command should be used only for handling uncommon configurations. Anything that you commonly configure should still be handled by using the specific configuration commands. The following is a list of the current patch variables documented and supported for this release.

Syntax: patch   bgp-subnets
                    dls-ignore-lfs
                    ethernet-security
                    filter-nr
                    hdx-delay
                    ip-default-ttl
                    ip-mtu
                    more-lines
                    mosheap-lowmark
                    ospf-import-rate
                    ping-size
                    ping-ttl
                    relax-jate
                    rip-static-suppress

Example: `patch ping-size`
                    `New value [0]?`

bgp-subnets new value
    If you want the BGP speaker to advertise subnet routes to its neighbors, set *new value* to 1. The default is 0.

    New value   The new value for the variable that you are patching.

dls-ignore-lfs
When set to 1, DLSw ignores the "largest frame" size bits in source-routed frames when setting up a circuit. This avoids circuit setup problems with some older LAN products that do not set these bits correctly. The default is 0.

ethernet-security
When set to a non-zero value, zeros the padding that is applied to Ethernet packets whose data portion is less than the physical minimum of 60 bytes. This may be required for security reasons. Default: 0.

filter-nr
Allows the NetBIOS "Name Recognized" to be filtered along with the current list of NetBIOS frames filtered by bridge code. NetBIOS Name filters will pass all NETBIOS packets that are not one of the following types: ADD_GROUP_NAME_QUERY, ADD_NAME_QUERY, DATAGRAM, NAME_QUERY. This parameter adds NAME_RECOGNIZED to the list of types.

hdx-delay
Enables the I_WFC feature of serial device drivers which delays I/O completion when interface is used in half-duplex mode.

ip-default-ttl
The TTL used in packets that are originated by the router. The default is 64.

> **Note:** It is preferable to set this parameter with the **set ttl** IP configuration command. (See the "Set" section of the "Using and Configuring IP" chapter of *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 1 Release 1*.) This patch variable remains for compatibility with configurations from older releases.

ip-mtu
This parameter limits the IP MTU size to the specified value. When this parameter is set, the IP MTU size on a given network interface is set to the lesser of the ip-mtu value and the largest value that network interface's configured frame size can accommodate.

more-lines
The number of lines to display on the console when listing the IP routing table, which uses a "more pipe" (¦).

mosheap-lowmark
This parameter specifies the percentage of free MOS heap memory, at which the device notifies the operator that an out-of-memory error is imminent. This notification allows the operator to take action to free up MOS heap memory before the device receives an error and stops.

When the operator receives notification, the operator can reconfigure the router and then reboot, minimizing the outage to the network. Specifying 0 for this parameter suppresses this warning.

Valid Values: 0 to 100

Default Value: 10

ospf-import-rate
Number of routes imported per second.

ping-size

The size of the data portion (that is, excluding IP and ICMP headers) of the ICMP ping packet that is sent via the IP>**ping** command. Default: 56 bytes. (The size of the ping data can also be entered as a parameter of the **ping** command as described in the "Ping" section of the "Monitoring IP" chapter of *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 1 Release 1*.)

ping-ttl

The TTL (time-to-live) sent in pings by the IP>**ping** command. Default: 64. (The TTL can also be entered as a parameter of the **ping** command as described in the "Ping" section of the "Monitoring IP" chapter of *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 1 Release 1*.)

relax-jate

Relaxes JATE ISDN restriction.

rip-static-suppress

When set to a non-zero value, static routes will not be advertised by RIP over a given interface unless the IP config> **enable send static** command is given for the interface. This changes the semantics of the **enable send static** command. When rip-static-suppress is equal to 0 (the default), the list of the routes advertised via RIP is the union of those specified by the interface's RIP flags.

Note: You must specify the complete name of the patch variable that you want to change. You cannot use an abbreviated syntax for the patch name.

Protocol

Use the **protocol** command at the Config> prompt to enter the configuration environment for the protocol software installed in the router. The **protocol** command followed by the desired protocol number *or* short name lets you enter a protocol's command environment. After you enter this command, the prompt of the specified protocol appears. From the prompt, you can enter commands specific to that protocol. Table 1-2 on page 1-14 lists examples of protocol numbers and names. To return to Config>, enter the **exit** command.

Notes:

1. To see the names and numbers of the protocols in your software load, at the Config> prompt, enter **list configuration**.

2. When you change a user-configurable parameter, you must restart the router for the change to take effect. To do so, enter the **reload** command at the OPCON prompt (*).

   The changes you make through CONFIG are kept in a configuration database in non-volatile memory and are recalled when you restart the router.

Syntax: protocol *prot#*

Example: **protocol 7**

                          or

                     protocol ipx

                     IPX config>

Qconfig

Use the **qconfig** command to initiate Quick Config. Quick Config allows you to configure parameters for bridging protocols, and routing protocols without entering separate configuration environments.

Syntax: qconfig

Note: For complete information on using the Quick Config software provided with your router, refer to the *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*.

Set

Use the **set** command to configure various system-wide parameters.

Syntax: set   contact-person . . .
                                       data-link . . .
                                       down-notify . . .
                                       global-buffers
                                       hostname
                                       inactivity-timer
                                       input-low-water
                                       location . . .
                                       packet-size
                                       prompt-level
                                       receive-buffers

contact-person *sysContact*
> Sets the name or identification of the contact person for this managed SNMP node. There is a limit of 80 characters for the *sysContact* name length.
>
> This variable is for information purposes only and has no effect on router operation. It is useful for SNMP management identification of the system.
>
> **Example:**   `set contact-person nautilus`

data-link   *type*   *interface#*
> Select the data link type for a serial interface or a dial circuit interface. The type can be one of:
>
> - FRAME-RELAY
> - PPP
> - SDLC
> - SRLY
> - V25BIS
> - X25
>
> **Notes:**
>
> 1. PPP and Frame Relay are the only data-links supported on dial circuit interfaces.
>
> 2. All data-link types can be used on the 8-port EIA 232E adapter, 6-port V.35/V.36 adapter, and the 8-port X.21 adapter except for V.25bis which can only be used with the EIA 232E adapter.

Interface# is the number of the interface you are configuring.

Example: `set data-link PPP 3`

down-notify   *interface# # of seconds*
>   Allows the user to specify the number of seconds before declaring an
>   interface as being down. The normal maintenance packet interval is 3
>   seconds, and it takes four maintenance failures to declare the interface as
>   down.
>
>   The **set down-notify** command is used primarily when tunneling LLC traffic
>   over an IP network using OSPF. If an interface goes down, OSPF cannot
>   detect it fast enough because of the length of time that it takes for an
>   interface to be declared down. Therefore, LLC sessions would begin to
>   timeout. You can set the down-notify timer to a lower value, allowing OSPF
>   to sense that an interface is down quicker. This enables an alternate route to
>   be chosen more quickly, which will prevent the LLC sessions from timing out.
>
>   **Note:** If the **set down-notify** command is executed on one end of a serial
>   link, the same command must be performed at the other end of the link or the
>   link may not come up and stay up.
>
>   *Interface#*
>   >   The number of the interface you are configuring.
>
>   *# of seconds*
>   >   The down notification time value that specifies the maximum time that
>   >   will elapse before a down interface is marked as such. Large values will
>   >   cause the router to ignore transient connection problems, and smaller
>   >   values will cause the router to react more quickly. The range of values
>   >   is 1 to 300 seconds and the default is 0, which sets the 3-second
>   >   period. Setting the down notification time to 0 will restore the default
>   >   time for that interface.
>
>   The **list devices** command will show the down notification time setting for any
>   interface that has the default value overridden.

Example: `set down-notify 4 3`

global-buffers *max#*
>   Sets the maximum number of global packet buffers, which are the packet
>   buffers used for locally originated packets. The default is to autoconfigure for
>   the maximum number of buffers (up to 1000). To restore the default, set the
>   value to 0. To display the setting for global-buffers, use the **list
>   configuration** command.

Example: `set global-buffers 30`

hostname
>   Adds or changes the router name. The router name is for identification only;
>   it does not affect any router addresses. The name must be:
>
>   - Less than 78 characters and is case sensitive

Example: `set hostname sales`

inactivity-timer *# of min*

> Changes the setting of the Inactivity Timer. The Inactivity Timer logs out a user if the remote or physical console is inactive for the period of time specified in this command. This command affect only consoles that require login. The default setting of 0 turns the inactivity timer off, indicating that no logoff is performed, no matter how long a console remains inactive.

> **Example:** `set inactivity-timer 3`

input-low-water *interface #   low # of receive buffers*

> Allows you to configure the value of the low number of receive buffers, or packets, on a per-interface basis, thus overriding the default values.

> The memory allocation strategy changes to conserve buffers when the number of free buffers is equal or less than the low or low-water mark value. When a packet is received, and the current value of the interface is less than the low water value, then that packet is eligible for flow control (dropping).

> The range of values is 1 to 255. The default is both platform and device specific. Setting the value to 0 restores the autoconfigured default.

> *Interface #* is the number of the interface you are configuring. *Low # of receive buffers* is the low water value.

> Lowering the value will make it less likely that packets from this interface will be dropped when sent on congested networks. However, lowering the value may negatively affect performance if it drops packets to the extent that the receive queue is frequently empty. Raising the value has the opposite effect.

> Type the **QUEUE** or **BUFFER** command at the GWCON prompt (**+**) to show the low setting.

> **Example:** `set input-low-water 4 7`

location *sysLocation*

> Sets the physical location of an SNMP node. There is a limit of 80 characters for the *sysLocation* name length. This variable is for information purposes only and has no effect on router operation. It is useful for SNMP management identification of the system.

> **Example:** `set location atlanta`

packet-size   *max packet size in bytes*

> Establishes or changes the maximum size of a packet buffer.

> **Attention:**   Use this command only under direct instructions from your service representative. **Never** use it to reduce packet size – **only** to increase it.

> **Example:** `set packet-size`

prompt-level *user-defined-name*

> Adds a user-defined name as a prefix to all operator prompts, replacing the hostname.

> The user-defined-name can be any combination of characters, numbers, and spaces up to 80 characters. Special characters may be used to request additional functions as described in Table 3-5 on page 3-32.

Example: `set prompt`
`What is the new MOS prompt [y]?` **`AnyHost 99`**
`AnyHost 99 Config>`

Table   3-5. Additional Functions Provided by the Set Prompt Level Command

Special Characters	Function Provided by the Set Prompt Level Command
$n	Displays the hostname. This is useful when you want the hostname included in the prompt. For example: `Config>` **`set prompt`** `What is the new MOS prompt [y]?` **`$n`** `hostname:: Config>`

$t	Displays the time. For example:
	
	`Config>` **`set prompt`**`.`
	`What is the new MOS prompt [y]?` **`$t`**
	`02:51:08[GMT-300] Config>`

$d	Displays the current date-month-year. For example:
	
	`Config>` **`set prompt`**`.`
	`What is the new MOS prompt [y]?` **`$d`**
	`26-Feb-1997 Config>`

$v	Displays the software VPD information in the following format: `program-product-number Feature xxxx Vx Rx.x PTFx RPQx`
$e	Erases one character *after* this combination within the user-defined prompt.
$h	Erases one character *before* this combination within the user-defined prompt.
$_	Adds a carriage return to the user-defined prompt.
$$	Displays the $.

Note: You can combine these commands. For example:

`Config>` **`set prompt`**`.`
`What is the new MOS prompt [y]?` **`$n::$d`**
`hostname::26-Feb-1997 Config>`

receive-buffers   *interface #   max #*
> Adjusts the number of private receive buffers for most interfaces. The range is 5 to 255.

> **Note:** This command is not applicable for ISDN Primary Rate Interfaces. For ISDN PRI, the interface handler determines the value based on the number of dial circuits configured.

> (On some devices, the maximum value is restricted further, as shown in Table 3-6 on page 3-33.) To restore the default, set the value to 0. The **set receive-buffers** command can be used to increase the receive performance of an interface. In addition, this command can be used to reduce flow control drops when the router is forwarding many packets from a fast interface to a slow interface. The effect of this command is visible on the GWCON **buffer** command.

> **Attention:**   Use this command only under direct instructions from your service representative.

> **Example:**   `set receive-buffers 4 30`

Table 3-6. Default and Maximum Settings for Interfaces

Interface	Default	Maximum
ATM	80	80
ETH	40	100
Serial	24	60
TKR	40	100

Time

Use the **time** command to set the 2216 system clock and date, and to display the values on the user console. These values can then be used to time-stamp ELS messages.

Note: The 2216 has a hardware clock that maintains the date and time after router re-initialization.

Syntax: t̲ime   host . . .
                                    list
                                    offset
                                    set . . .
                                    sync . . .

host *IP_address*
> Sets the IP address of the RFC 868-compliant host which will be used as the time source. This is the address of a host which will respond to an empty datagram on UDP port 37 with a datagram containing the current time.
>
> **Example:**   **time host 131.210.1.4**

list
> Displays all configured time-related parameters. This includes the current time (if set) and the source of the time (operator or IP address from which time was last received).
>
> **Example:**   **time list**
>
> ```
> 05:20:27  Wednesday December 7, 1994
> Set by: operator
> Time Host:  131.210.4.1          Sync Interval: 10
> seconds
> GMT Offset:  -300 minutes
> ```

offset *minutes*
> Defines the time zone, in minutes, offset from GMT (Greenwich Mean Time). Note that values west of GMT are negative. For example, EST is 5 hours earlier than GMT, so the command would be **time offset -300**.
>
> **Example:**   **time offset**
>
> ```
> minutes from GMT (-720 to 720) [0]? -300
> ```

set *<year month date hour minute second>*
> Prompts you to set the current time. If you do not specify the entire time in the command, you are prompted for the remaining values. You can change the date as shown in the following example.

Example: `time set`

```
year [1996] 1997
month [12]?
date [6]? 7
hour [11]? 12
minute [3]?
second [2]?
```

sync *seconds*
> Sets the period, in seconds, at which the router will poll the time host for the current time.

Example: `time sync 300`

Unpatch

Use the **unpatch** command to restore the values of the patch variables entered with the **patch** command to their default values. See the **patch** command in "Patch" on page 3-26.

Note: You *must* specify the long name of the patch variable to be restored.

Syntax: <u>unpatch</u> *variable name*

Example: `unpatch ethernet-security`

Update

Use the **update** command to update the configuration memory when you receive a new software load. Follow the instructions on the release notice sent with the software. The **update** command is the last command that you enter when loading new software. After you enter this command, the console displays a message indicating configuration memory is being updated.

Syntax: <u>u</u>pdate <u>v</u>ersion-of-SRAM

Example: `update version-of-SRAM`

`Updating configuration memory to V15.2 [X104]`

Chapter 4. Performing Change Management Operations

This chapter describes how to use the 2216 change management functions. This chapter includes the following sections:

- "Understanding Change Management"
- "Using the Trivial File Transfer Protocol (TFTP)"
- "Entering and Exiting Change Management Configuration Environment" on page 4-2
- "Change Management Configuration Commands" on page 4-2

Understanding Change Management

Change management refers to the handling of software and configuration data for a 2216. This involves:

1. Moving code and configuration data to and from the 2216

2. Moving code and configuration data on the 2216 persistent storage device, which is currently a disk drive and flash memory

3. Selecting and Activating specific combinations of software and configuration.

The change management functions are available through the `Boot config>` environment (available through talk 6), or the firmware should the box be in a condition where the hard drive does not contain viable software (that is, you cannot access talk 6). During normal operations, access the change management functions through the `Boot config>` prompt.

The 2216 code and configuration data storage resource is divided into areas called "system banks" (banks for short), each containing a single version of the operational code and any other ancillary files pertinent to that release of the code. Up to four configuration files are associated with each bank's software.

The general change management model of the 2216 is to introduce new code and or configuration data to the system while the system runs at its present level and then activate the changed code or configuration data set later. If for some reason the new code or configuration does not function as expected, you have the ability to revert to the previous version of the configuration.

Using the Trivial File Transfer Protocol (TFTP)

TFTP is a file transfer protocol that runs over the Internet UDP protocol. This implementation provides multiple, simultaneous TFTP file transfers between a 2216's non-volatile configuration memory, image bank, and remote hosts.

TFTP allows you to:

- Get a configuration file from a server to a 2216
- Put a configuration file from a 2216 to a server

TFTP transfers involve a *client* node and a *server* node. The client node generates a TFTP Get or Put request onto the network. The 2216 acts as a client node by

generating TFTP requests from the 2216 console using the `Boot config>` process **tftp** command.

The client can transfer a copy of a configuration file or image file stored in an image bank of a server.

The server is any device (for example, a personal computer or workstation) that receives and services the TFTP requests. When the 2216 acts as a server, transfers are transparent to the user. Use the ELS subsystem TFTP message log to view the transfer in progress.

Entering and Exiting Change Management Configuration Environment

To enter the change management configuration command environment, use the CONFIG **boot** command. When the router's software is initially loaded, it is running in the OPCON process, signified by the * prompt. From the * prompt:

1. Enter **talk 6**.

2. At the `Config>` prompt, type **boot**.

3. At the `Boot config>` prompt, type **?**. The following appears:

```
ADD description
COPY software
DESCRIBE software
ERASE software
LIST software
LOCK Config file
SET boot information
TFTP software
UNLOCK Config file
EXIT
```

To return to the CONFIG process, type **exit**.

Change Management Configuration Commands

This section describes the Change Management Configuration commands. Each command includes a description, syntax requirements, and an example. Table 4-1 on page 4-3 summarizes the Change Management Configuration commands.

After accessing the Change Management Configuration environment, enter the configuration commands at the `Boot config>` prompt.

Table   4-1. Change Management Configuration Commands

Command	Function
? (Help)	Displays a list of the commands available from that prompt level.
Add	Adds an optional description to a configuration file.
Copy	Copies boot files and configuration files to or from banks.
Describe	Displays information about the stored loadfile images.
Erase	Erases a stored image or a configuration file.
List	Displays information about load images and configuration files.
Lock	Prevents the device from overwriting the selected configuration with any other configuration.
Set	Selects code bank and configuration to be used.
TFTP	Initiates TFTP file transfers between 2216 remote servers.
Unlock	Removes the lock from a configuration allowing the configuration to be updated by the device.
Exit	Leaves the Change Management Configuration environment and returns to the CONFIG process.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: Boot config>**?**

```
ADD description
COPY software
DESCRIBE software VPD
ERASE software
LIST software status
LOCK Config File
SET boot information
TFTP software
UNLOCK Config File
EXIT
```

Add

Use the **add** command to add an optional description to a configuration file.

Syntax: <u>a</u>dd   configuration *file* load *image description*

Example: Boot config> **add**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 01:26 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 01:39 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 01:52 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970 00:30 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config      L - Config File is Locked


Select the source bank:  (A, B): [A]
Select the source configuration:  (1, 2, 3, 4): [1] 3
Enter the description of the file: () New config for today

Attempting to set description for bank A configuration 3.

Operation completed successfully.


Boot config>list
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               | 01 Jan 1970       |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:58 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        | New config for today          | 09 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 01:05 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970       |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config      L - Config File is Locked
```

COPY

Use the **copy** command to copy configuration files and image files to and from image banks.

Syntax: <u>co</u>*py*   configuration [file]

                load [image]

Example: `Boot config>`**`copy load`**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - AVAIL          |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 01:26 |
| CONFIG 2 - AVAIL    *  | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 01:39 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 01:52 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:37 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-----------------------------+------------------+
  * - Last Used Config     L - Config File is Locked


Select the source bank:  (A, B): [A] b
Select the destination bank:  (A, B): [B] a
Copy SW load image from: bank B
                  to: bank A.


Operation completed successfully.
```

Example: `Boot config>`**`copy configuration`**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - CORRUPT        |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 01:26 |
| CONFIG 2 - AVAIL    *  | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 01:39 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 01:52 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:37 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-----------------------------+------------------+
  * - Last Used Config     L - Config File is Locked


Select the source bank:  (A, B): [A]
Select the source configuration:  (1, 2, 3, 4): [1]
Select the destination bank:  (A, B): [B]
Select the destination configuration:  (1, 2, 3, 4): [1]
Copy SW configuration from: bank A, configuration 1
                     to: bank B, configuration 1.
/hd0/sys0/CONFIG0 --> /hd0/sys1/CONFIG0

Operation completed successfully.
```

If the copy fails you may receive one of the following messages:

Error: Active bank cannot be overwritten or erased.
    You attempted to copy a configuration into the image currently in use by the 2216.

Error: File copy failed.
This condition occurs when the copy operation fails for reasons other than copying to the active configuration. The most common cause is specifying the same source and destination configurations. When you list (see "LIST" on page 4-9) the configurations, CORRUPT appears next to the bank that is damaged.

DESCRIBE command

Use the **describe** command to display information about a stored image.

Syntax: <u>de</u>scribe

Example: `Boot config>`**`describe`**

```
+------------------------+------------------------+
|          BANK A        |          BANK B        |
| Product ID -   5765-B87 | Product ID -   5765-B87 |
| Version   1 Release   1 | Version   1 Release   1 |
| Maint.    0 PTF       0 | Maint.    0 PTF       0 |
| Feat.  2801 RPQ       0 | Feat.  2801 RPQ       0 |
| Date        31 Dec 1996 | Date        31 Dec 1996 |
+------------------------+------------------------+
```

ERASE command

Use the **erase** command to erase a stored image or a configuration file.

Syntax: <u>e</u>*rase*   configuration [file]

load [image]

config *or* load
Erases a configuration image or a boot image. Enter the config or the boot image number that contains the load after the **erase** command.

Example: `Boot config>`**`erase load`**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - CORRUPT        |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 01:26 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 00:39 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config     L - Config File is Locked


Select the bank to erase:  (A, B): [A] a
Erase SW load image from bank A.

Operation completed successfully.


Boot config>list
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               | 01 Jan 1970       |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:26 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 00:39 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970       |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config     L - Config File is Locked
```

Example: Boot config>**erase configuration**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                                | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:26 |
| CONFIG 2 - AVAIL   *   | test config for pubs           | 01 Jan 1970 01:13 |
| CONFIG 3 - AVAIL       |                                | 01 Jan 1970 01:26 |
| CONFIG 4 - AVAIL       |                                | 01 Jan 1970 01:39 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                                | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                                | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                                | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                                | 01 Jan 1970 00:24 |
+-----------------------+------------------------------+------------------+
  * - Last Used Config      L - Config File is Locked


Select the source bank:  (A, B): [A]
Select the configuration to erase:  (1, 2, 3, 4): [1] 3
Erase SW configuration file from bank A, configuration 3.

Operation completed successfully.


Boot config>list
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                                |                   |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:14 |
| CONFIG 2 - AVAIL   *   | test config for pubs           | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                                | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                                | 01 Jan 1970 00:26 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                                | 01 Jan 1970         |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                                | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                                | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                                | 01 Jan 1970 00:24 |
+-----------------------+------------------------------+------------------+
  * - Last Used Config      L - Config File is Locked
```

Notice that the list command displays *NONE* by bank A, config 3.

If the erasure fails, a message indicating the failure appears on the console along with the banks that failed.

LIST

Use the **list** command to display information about which load images and configuration files are available and active.

Syntax: list

Example: Boot config>**list**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - AVAIL          |                                | 01 Jan 1970       |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:26 |
| CONFIG 2 - AVAIL   *   | test config for pubs           | 01 Jan 1970 01:13 |
| CONFIG 3 - AVAIL       |                                | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                                | 01 Jan 1970 00:39 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                                | 01 Jan 1970       |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                                | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                                | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                                | 01 Jan 1970 00:24 |
+-----------------------+-----------------------------+------------------+
  * - Last Used Config    L - Config File is Locked
```

The following are the possible file status descriptors:

ACTIVE

The file is currently loaded and is running on the 2216

AVAIL

This is a valid file that can be made ACTIVE.

CORRUPT

The file was damaged or not loaded into the 2216 completely. The file must be replaced.

LOCAL

The file will be used only on the next reload or reset. After the file is used, it will be placed in AVAIL state.

PENDING

This file will be loaded on the next reload, reset, or power-up of the 2216.

LOCK

Use the **lock** command to prevent the device from overwriting the selected configuration with any other configuration.

Syntax: lock

Example: Boot config>**lock**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE          |                               | 01 Jan 1970 01:03 |
| CONFIG 1 - AVAIL      | test config for pubs          | 01 Jan 1970 00:26 |
| CONFIG 2 - AVAIL   *  | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE       |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL      |                               | 01 Jan 1970 00:26 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE        |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL      | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL      |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL      |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *  |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config     L - Config File is Locked


Select the source bank:  (A, B): [A]
Select the source configuration:  (1, 2, 3, 4): [1] 4
Attempting to lock bank A and configuration 4.

Operation completed successfully.


Boot config>list
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE          |                               | 01 Jan 1970       |
| CONFIG 1 - AVAIL      | test config for pubs          | 01 Jan 1970 00:13 |
| CONFIG 2 - AVAIL   *  | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE       |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL    L |                               | 01 Jan 1970 00:26 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE        |                               |                   |
| CONFIG 1 - AVAIL      | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL      |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL      |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *  |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config     L - Config File is Locked
```

SET command

Use the **set** command to select the code bank and the configuration to use and the duration of use. The valid durations are:

once   The configuration is active for the next boot only.

always The configuration is active for all subsequent boots until changed again.

Syntax: s̲et

Example: Boot config>**set**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               | 01 Jan 1970 01:03 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:13 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 00:26 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config      L - Config File is Locked


Select the source bank:  (A, B): [A] b
Select the source configuration:  (1, 2, 3, 4): [1] 4
Select the duration to use for booting:  (once, always): [always]
Set SW to boot using bank B and configuration 4, always.

Operation completed successfully.


Boot config>list
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               | 01 Jan 1970        |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:13 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 00:26 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970        |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config      L - Config File is Locked
```

TFTP command

Use the **tftp** command to initiate TFTP file transfers between 2216 remote servers.

Syntax:

```
tftp    get   config
              load

        put   config
              load
```

Example: Boot config>**tftp get load**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                                | 01 Jan 1970 01:03 |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:01 |
| CONFIG 2 - AVAIL   *   | test config for pubs           | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                                | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                                | 01 Jan 1970 00:14 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                                | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs           | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                                | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                                | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                                | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+------------------+
 * - Last Used Config      L - Config File is Locked


Specify the server IP address (dotted decimal): : [1.2.3.4] 192.9.200.1
Specify the remote file name: : (load.img) /usr/2216load/c200-rtr.img
Select the destination bank:  (A, B): [A] a
TFTP SW load image
  get:   /usr/2216load/c200-rtr.img
  from:  192.9.200.1
  to:    bank A.

Operation completed successfully.
```

UNLOCK

Use the **unlock** command to allow the device to overwrite the selected configuration that was previously locked.

Syntax: unlock

Example: Boot config>**unlock**

```
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               | 01 Jan 1970 01:03 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:13 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL    L  |                               | 01 Jan 1970 00:26 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970 00:01 |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config      L - Config File is Locked


Select the source bank:  (A, B): [A]
Select the source configuration:  (1, 2, 3, 4): [1] 4
Attempting to unlock bank A and configuration 4.

Operation completed successfully.
Boot config>list
+------ BankA -----------+--------- Description ----------+------ Date -------+
| IMAGE - NONE           |                               |                   |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:01 |
| CONFIG 2 - AVAIL   *   | test config for pubs          | 01 Jan 1970 01:13 |
| CONFIG 3 - NONE        |                               | 01 Jan 1970 00:58 |
| CONFIG 4 - AVAIL       |                               | 01 Jan 1970 00:14 |
+------ BankB -----------+--------- Description ----------+------ Date -------+
| IMAGE - ACTIVE         |                               | 01 Jan 1970         |
| CONFIG 1 - AVAIL       | test config for pubs          | 01 Jan 1970 00:54 |
| CONFIG 2 - AVAIL       |                               | 01 Jan 1970 00:01 |
| CONFIG 3 - AVAIL       |                               | 01 Jan 1970 00:14 |
| CONFIG 4 - ACTIVE  *   |                               | 01 Jan 1970 00:24 |
+-----------------------+-------------------------------+-------------------+
  * - Last Used Config      L - Config File is Locked
```

Exit

Use the **exit** command to leave the current process command level and return to the previous process level.

Syntax: exit

Example: exit

Chapter 5. Accessing the Firmware from the Command Line Interface

This section covers boot options that can be set from the `Firmware and Operational` command prompt. For information about file transfer and file management, refer to *IBM 2216 Nways Multiaccess Connector Hardware Installation Guide*.

The 2216 is designed to boot from one of the integrated image banks. With the hard drive installed, the 2216 has two image banks that can be used to IML or boot the device. The 2216 also has options to come up in Attended or Unattended mode. Attended mode requires direct interaction with a user at a console attached on the serial port.

The device can be booted from the hard drive, known as Bank A and B.

Accessing the Firmware Prompt

Before booting the 2216, note the following:

- You will need a terminal or IP workstation connected to the 2216. This can be a VT100 TTY device connected directly through the serial port. You can connect an IP workstation using SLIP to connect into the 2216. The default 2216 IP address is 10.1.1.2 and the workstation address should be 10.1.1.3.

Important: To access the Firmware prompt, you can stop the 2216 boot; to stop it, you must have a TTY console directly attached to the serial port. When the 2216 starts its boot sequence, press **Ctrl** **C** from the console to interrupt the boot sequence.

Another way to control booting is to configure the 2216 to come up in Attended mode. Attended mode can be configured from the Firmware command set.

Boot Options Available for the 2216

The 2216 can be configured for Unattended mode. In Unattended mode, you must have chosen which load image and which configuration to load. You are provided with two banks to choose among. The structure of the image banks is as follows:

- IMAGE - Status of image

- CONFIG 1 - Status of Config

- CONFIG 2 - Status of Config

- CONFIG 3 - Status of Config

- CONFIG 4 - Status of Config

See "LIST" on page 4-9 for a description of file statuses.

Attended Mode

When the 2216 is configured to come up in Attended mode, you are given access to the Firmware command set. From this level of commands, you can select the Image Bank from which to load and the config. You can at this point load new config files or image files. This connection is either a TTY or Telnet connection. You can transfer files using the Xmodem protocol for TTY or TFTP for IP connections.

In Attended mode, you can start booting the 2216 by pressing **F9** or **<Esc>9** to start the operating system.

Unattended Mode

This is the normal mode for the 2216. It will come up on the Active, Local, or Pending image and config based on your choice.

Chapter 6. The GWCON (Monitoring) Process and Commands

This chapter describes the GWCON process and includes the following sections:

- "What is GWCON?"
- "Entering and Exiting GWCON" on page 6-2
- "GWCON Commands" on page 6-2

What is GWCON?

The Gateway Console (monitoring) process, GWCON (also referred to as CGWCON), is a second-level process of the router user interface.

Using GWCON commands, you can:

- List the protocols and interfaces currently configured in the router.
- Display memory and network statistics.
- Set current Event Logging System (ELS) parameters.
- Test a specified network interface.
- Communicate with third-level processes, including protocol environments.
- Enable and disable interfaces.

GWCON fits into the router software structure as shown in Figure 6-1.

Router Software Processes

```
                    ┌──────────────┐
                    │    OPCON     │
                    │      *       │
                    └──────┬───────┘
        ┌───────────┬──────┴──────┬───────────┐
   talk 6       talk 5                      talk 2
   ┌────────┐  ┌────────┐  ┌────────┐  ┌────────┐
   │ config │  │ GWCON  │  │ MOSDBG │  │ MONITR │
   │CONFIG >│  │   +    │  │        │  │        │
   └────────┘  └────────┘  └────────┘  └───┬────┘
                                      ┌─────┴──────┐
                                      │ELS Messages│
                                      └────────────┘
```
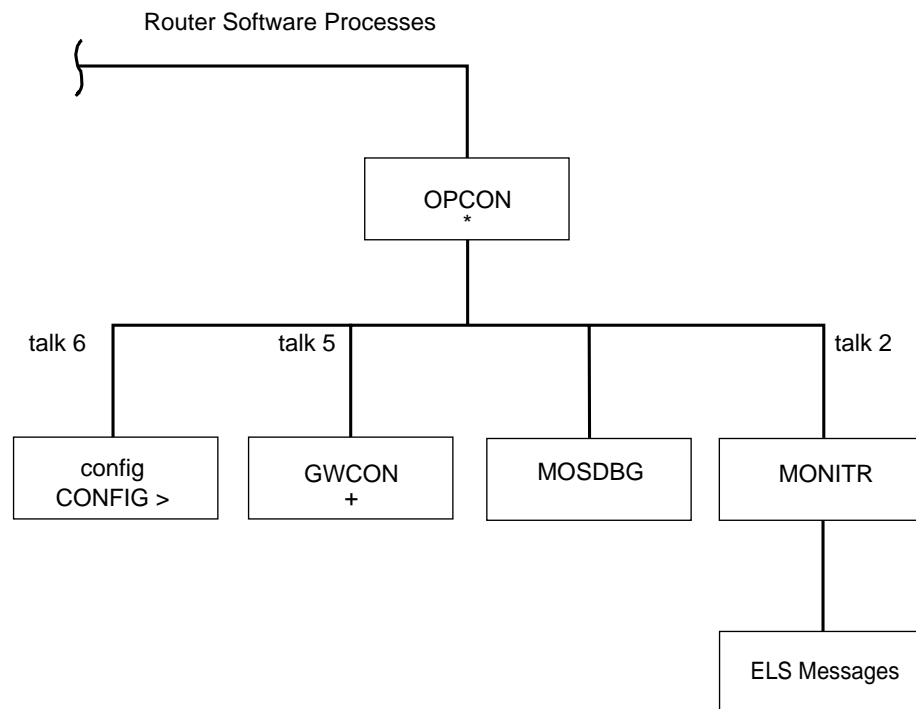
Figure   6-1. GWCON in the Router Software Structure

The GWCON command interface is made up of levels called modes. Each mode has its own prompt. For example, the prompt for the IP protocol is IP>.

If you want to know the process and mode you are communicating with, press **Return** to display the prompt. Some commands in this chapter, such as the **network** and **protocol** commands, allow you to access the various modes in GWCON.

Entering and Exiting GWCON

To enter the GWCON command environment from OPCON and obtain the GWCON prompt:

1. At the OPCON prompt, enter the **status** command to find the process ID (PID) of GWCON.

   ` * status`

   For sample output of the status command see "Status" on page 2-8.

2. Enter the **talk** command and the PID for GWCON to get to the GWCON prompt.

   ` * talk 5`

The console displays the GWCON prompt (+). If the prompt does not appear, press **Return**. Now, you can enter GWCON commands.

To return to OPCON, enter the OPCON intercept character. (The default is `Ctrl` `P` .)

GWCON Commands

This section contains the GWCON commands. Each command includes a description, syntax requirements, and an example. The GWCON commands are summarized in Table 6-1.

To use the GWCON commands, access the GWCON process by entering **talk 5** and enter the GWCON commands at the (+) prompt.

Table 6-1 (Page 1 of 2). GWCON Command Summary	
Command	**Function**
? (Help)	Lists the GWCON commands.
Buffer	Displays information about packet buffers assigned to each interface.
Clear	Clears network statistics.
Configuration	Lists status of the current protocols and interfaces.
Disable	Takes the specified interface off line.
Error	Displays error counts.
Event	Enters the Event Logging System environment.
Feature	Provides access to console commands for independent router features outside the usual protocol and network interface console processes.
Interface	Displays network hardware statistics or statistics for the specified interface.
Memory	Displays memory, buffer, and packet data.

Table 6-1 (Page 2 of 2). GWCON Command Summary

Command	Function
Network	Enters the console environment of the specified network.
Protocol	Enters the command environment of the specified protocol.
Queue	Displays buffer statistics for a specified interface.
Statistics	Displays statistics for a specified interface.
Test	Enables a disabled interface or tests the specified interface.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax:	?
Example:	**?**

```
BUFFER statistics
CLEAR statistics
CONFIGURATION of gateway
DISABLE interface
ERROR counts
EVENT logging
FEATURE commands
INTERFACE statistics
MEMORY statistics
NETWORK commands
PROTOCOL commands
QUEUE lengths
STATISTICS of network
TEST network
```

Example:	**protocol ?**

```
IP
ARP
DNA
VINES
IPX
OSI
DVMRP
BGP
SNMP
OSPF
SDLC Relay
AP2
ASRT
HST
LNM
DLSW
XTP
```

Buffer

Use the **buffer** command to display information about packet buffers assigned to each interface.

Note: Each buffer on a device is the same size and is dynamically built. Buffers vary in size from one device to another.

To display information about one interface only, enter the interface or network number as part of the command. To obtain the interface number, use the GWCON **configuration** command.

Syntax: bu̲ffer *network#*

Example: buffer

```
           Input  Buffers:        Buffer sizes:
Nt  Interface  Req  Alloc  Low  Curr  Hdr  Wrap  Data  Trail  Total  Bytes Alloc
0   TKR/0      20   20     7    0     109  92    2052  7      2260   45200
1   PPP/0      20   20     7    20    109  92    2052  7      2260   45200
2   PPP/1      10   10     4    0     108  92    2048  0      2248   22480
```

Nt	Network interface number associated with the software.
Interface	Type of interface.

Input Buffers:

Req	Number of buffers requested.
Alloc	Number of buffers allocated.
Low	Low water mark (flow control).
Curr	Current number of buffers on this device. The value will be 0 if the device is disabled. When a packet is received, if the value of *Curr* is below *Low*, then the packet is eligible for flow control. (See the **queue** command for conditions.)

Buffer Sizes:

Hdr	Sum of the maximum hardware, MAC, and data link headers.
Wrap	Allowance given for MAC, LLC, or Network layer headers due to protocol wrapping.
Data	Maximum data link layer packet size.
Trail	Sum of the largest MAC and hardware trailers.
Total	Overall size of each packet buffer.
Bytes Alloc	Amount of buffer memory for this device. This value is determined by multiplying the values of *Alloc x Total*.

Clear

Use the **clear** command to delete statistical information about one or all of the router's network interfaces. This command is useful when tracking changes in large counters. Using this command does not save space or speed up the router.

Enter the interface (or net) number as part of the command. To get the interface number, use the GWCON **configuration** command.

Syntax: <u>c</u>lear *interface#*

Example:
```
clear 1
Clear network statistics? (Yes or No):__
```

Configuration

Use the **configuration** command to display information about the protocols and network interfaces. The output is displayed in three sections, the first section lists the router identification, software version, boot ROM version, and the state of the auto-boot switch. The second and third sections list the protocol and interface information.

Syntax: <u>co</u>nfiguration

Example: **configuration**

```
Multiprotocol Access Services

5765-B87 Feature 2802 V1 R1.0 PTF 0 RPQ 0



Num Name  Protocol
0   IP    DOD-IP
3   ARP   Address Resolution
4   DN    DNA Phase IV
6   VIN   Banyan Vines
7   IPX   NetWare IPX
10  BGP   Border Gateway Protocol
11  SNMP  Simple Network Management Protocol
12  OSPF  Open SPF-Based Routing Protocol
22  AP2   AppleTalk Phase 2
23  ASRT  Adaptive Source Routing Transparent Enhanced Bridge
26  DLS   Data Link Switching
27  XTP   X.25 Transport Protocol
28  APPN  Advanced Peer-to-Peer Networking [HPR]
30  APPN  Advanced Peer-to-Peer Networking [ISR]

Num Name  Feature
2   MCF   MAC Filtering

16 Networks:
Net Interface  MAC/Data-Link     Hardware          State
0   TKR/0      Token-Ring/802.5  Token-Ring        Up
1   TKR/1      Token-Ring/802.5  Token-Ring        Up
2   TKR/2      Token-Ring/802.5  Token-Ring        Up
3   TKR/3      Token-Ring/802.5  Token-Ring        Up
4   Eth/0      Ethernet/IEEE 802.3  Ethernet       Up
5   Eth/1      Ethernet/IEEE 802.3  Ethernet       Up
6   Eth/2      Ethernet/IEEE 802.3  Ethernet       Up
7   Eth/3      Ethernet/IEEE 802.3  Ethernet       Up
8   Eth/4      Ethernet/IEEE 802.3  Ethernet       Up
9   Eth/5      Ethernet/IEEE 802.3  Ethernet       Up
10  FR/0       Frame Relay       V.35/V.36         Up
11  X25/0      X.25              V.35/V.36         Up
12  PPP/0      Point to Point    V.35/V.36         Up
13  PPP/1      Point to Point    V.35/V.36         Up
14  PPP/2      Point to Point    V.35/V.36         Up
15  PPP/3      Point to Point    V.35/V.36         Up
```

- The first line lists the product name.

- The second line lists the program/product number, Feature Number, Version, Release, PTF and RPQ information.

- The configured protocols are listed, followed by the configured features.

Protocols:

  Num
         Number that is associated with the protocol.

  Name
         Abbreviated name of the protocol.

  Protocol
         Full name of the protocol.

Features:

Num
Number associated with the feature.

Name
Abbreviated name of the feature.

Feature
Full name of the feature.

Networks:

Net
Network number that the software assigns to the interface. Networks are numbered starting at 0. These numbers correspond to the interface numbers discussed under the CONFIG process.

Interface
Name of the interface and instance of this type of interface.

MAC/Data Link
Type of MAC/Data link configured for the interface.

Hardware
Specific kind of interface by hardware type.

State
Current state of the network interface.

Testing   Indicates that the interface is undergoing a self-test. Occurs when the router is first started, when a problem is detected on the interface, or when the **test command** is used. (The **enable slot** command can also be used to initiate a self-test of all interfaces on an adapter.)

When an interface is operational, the interface periodically sends out maintenance packets and/or checks the physical state of the port or line to ensure that the interface is still functioning properly. If the maintenance fails, the interface is declared down and a self-test is scheduled to run in 5 seconds. If a self-test fails, the interface transitions to the down state and the interval until the next self-test is increased up to a maximum of 2 minutes. If the self-test is successful, the network is declared up.

Up   Indicates the interface is operational.

Down   Indicates that the interface is not operational and has failed a self-test. The network will periodically transition to the testing state to determine if the interface can become operational again.

Disabled   Indicates that the interface is disabled. An interface can be disabled by the following methods:

- An interface can be configured as disabled using the CONFIG **disable** command. Each time that the router is re-initialized, the interface's initial state will be

disabled. It will remain in the disabled state until an action is taken to enable it.

- An interface can be disabled using the GWCON **disable** command. This method is temporary because the interface will revert to its configured state (enabled or disabled) when the router is re-initialized.

- The network manager can disable the interface through SNMP. This method is temporary because the interface will revert to its configured state (enabled or disabled) when the router is re-initialized.

When an interface is disabled, it remains disabled until one of the following methods is used to enable it:

- The GWCON **test** command is used to start a self-test of the interface.

- The GWCON **enable slot** command is used to start a self-test on all interfaces on an adapter.

- The network manager initiates a self-test of the interface through SNMP.

WAN Reroute also can change the state of a disabled interface. If an interface is configured as an alternate interface for WAN Reroute and its configured state is disabled, WAN Reroute will start a self-test of the interface when the primary interface goes down. When the primary interface is operational and stable again, WAN Reroute puts the alternate interface back to its configured state. Refer to Chapter 40, "The WAN Reroute Feature" on page 40-1 for more information.

Available	Indicates that the interface has been configured as a secondary WAN Restoral interface and it is available to back up the primary interface.
Not Present	Indicates that the interface's adapter is not plugged in or that the configured adapter type does not match the adapter type that is actually present in the slot.
	Not Present is also used as the state for a null device.
HW Failure	Indicates that there is an unrecoverable hardware error for the interface's hardware.
Diagnostics	Indicates that hardware diagnostics are running.

Disable

Use the **disable** command to take a network interface off-line, making the interface unavailable. This command immediately disables the interface. You are not prompted to confirm, and no verification message displays. If you disable an interface with this command, it remains disabled until you use the GWCON **test** command or an OPCON **restart** or **reload** command to enable it.

Enter the interface, or net number as part of the command. To obtain the interface number, use the GWCON **configuration** command.

Note: This command may not be accepted during interface self-testing. In this case, one of the following messages will be displayed:

```
Network is already undergoing test, cancelling further testing
Network is already undergoing test, cannot disable now
```

Try the **disable** command again later.

Syntax: <u>d</u>isable   *interface#*

Example: **disable 1**

Error

Use the **error** command to display error statistics for the network. This command provides a group of error counters.

Syntax: <u>e</u>rror

Example: **error**

```
                      Input   Input     Input      Input    Output  Output
   Nt  Interface   Discards  Errors  Unk Proto  Flow Drop  Discards  Errors
   0   TKR/0              0       0          0          0         0       0
   1   PPP/0              0       0          0          0         0       0
   2   PPP/1              0       0          0          0         0       0
```

Nt   Network interface number associated with the software.

Interface   Type of interface.

Input Discards   Number of inbound packets which were discarded even though no errors were detected to prevent their being deliverable to a higher-layer protocol. The packets may have been discarded to free buffer space.

Input Errors   Number of packets that were found to be defective at the data link.

Input Unk Proto Number of packets received for an unknown protocol.

Input Flow Drop Number of packets received that are flow controlled on output.

Output Discards Number of packets that the router chose to discard rather than transmit due to flow control.

Output Errors   Number of output errors, such as attempts to send over a network that is down or over a network that went down during transmission.

Note: The sum of the discarded output packets is not the same as input flow drops over all networks. Discarded output may indicate locally originated packets.

Event

Use the **event** command to access the Event Logging System (ELS) console environment. This environment is used to set up temporary message filters for troubleshooting purposes. All changes made in the ELS console environment will take effect immediately, but will go away when the router is reinitialized. See Chapter 8, "Using and Configuring the Event Logging System (ELS)" for information about the Event Logging System and its commands. Use the **exit** command to return to the GWCON process.

Syntax: <u>e</u>vent

Example: `event`

Feature

Use the **feature** command to access console commands for specific 2216 features outside of the protocol and network interface console processes.

Enter a question mark after the **feature** command to obtain a listing of the features available for your software release. For example:

`feature ?`

To access that feature's console prompt, enter the **feature** command at the GWCON prompt followed by the feature number or short name. Table 3-4 on page 3-23 lists available feature numbers and names.

Once you access the prompt for that feature, you can begin entering specific commands to monitor that feature. To return to the GWCON prompt, enter the **exit** command at the feature's console prompt.

Syntax: f̲eature   *feature# OR feature-short-name*

Example: `feature 1`
         or
       `feature brs`

       `Reservation console>`

Interface

Use the **interface** command to display statistical information about the network interfaces (for example, Ethernet or Token-Ring). This command can be used without a qualifier to provide a summary of all the interfaces (shown in the following output) or with a qualifier to reveal detailed information of one specific interface.

Descriptions of detailed output for each type of interface are provided in the specific interface *Monitoring* chapters found in this guide. To obtain the interface number, use the GWCON **configuration** command.

Syntax: i̲nterface   *interface#*

Example: `interface`

```
                                     Self-Test  Self-Test  Maintenance
Nt Nt'  Interface  Slot-Port         Passed     Failed     Failed
0  0    TKR/0      Slot: 1  Port: 1      1          0          0
1  1    TKR/1      Slot: 1  Port: 2      2          1          0
2  2    TKR/2      Slot: 2  Port: 1      2          1          0
3  3    TKR/3      Slot: 2  Port: 2      2          1          0
4  4    Eth/0      Slot: 4  Port: 1      1          0          0
5  5    Eth/1      Slot: 4  Port: 2      1          0          0
6  6    Eth/2      Slot: 5  Port: 1      1          0          0
7  7    Eth/3      Slot: 5  Port: 2      3          2          2
8  8    Eth/4      Slot: 6  Port: 1      1          0          0
9  9    Eth/5      Slot: 6  Port: 2      5          4          1
10 10   FR/0       Slot: 8  Port: 0      2          1          0
11 11   X25/0      Slot: 8  Port: 1      1          0          0
12 12   PPP/0      Slot: 8  Port: 2      2          1          0
13 13   PPP/1      Slot: 8  Port: 3      1          0          0
14 14   PPP/2      Slot: 8  Port: 4      1          0          0
15 15   PPP/3      Slot: 8  Port: 5      1          0          0
```

Note: The display varies depending on the device.

Nt	Global interface number.
Nt'	Reserved for dial circuit use. Interface number of the physical network interface that the dial circuit uses.
Interface	Interface name.
Slot-Port	Slot number and port number of the interface.
Self-Test Passed	Number of times self-test succeeded (state of interface changes from down to up).
Self-Test Failed	Number of times self-test failed (state of interface changes from up to down).
Maintenance Failed	Number of maintenance failures.

Memory

Use the **memory** command to display the current CPU memory usage in bytes, the number of buffers, and the packet sizes.

To use this command, free memory must be available. The number of free packet buffers may drop to zero, resulting in the loss of some incoming packets; however, this does not adversely affect router operations. The number of free buffers should remain constant when the router is idle. If it does not, contact your service representative.

Syntax: <u>m</u>emory

Example: **memory**

```
                Total  Reserve   Never      Perm     Temp     Prev
                                 Alloc      Alloc    Alloc    Alloc

Heap memory    5463895  201824  5065383    328344   375856    22656

Number of global buffers:  Total = 294, Free = 287, Fair = 57, Low = 58
Global buff size:  Data = 4478, Header = 128, Wrap = 92, Trailer = 19
Total = 4700
```

Heap memory:	Amount of memory used to dynamically allocate data structures.
Total	Total amount of space available for allocation for memory.
Reserve	Minimum amount of memory needed by the currently configured protocols and features.
Never Alloc	Memory that has never been allocated.
Perm Alloc	Memory requested permanently by router tasks.
Temp Alloc	Memory allocated temporarily to router tasks.
Prev Alloc	Memory allocated temporarily and returned.
Number of global buffers:	
Total	Total number of global buffers in the system.
Free	Number of global buffers available.
Fair	Fair number of buffers for each interface. (See Low below.)

Low	The number of free buffers at which the allocation strategy changes to conserve buffers. If the value of *Free* is less than *Low*, then buffers will not be placed on any queue that has more than the *Fair* number of buffers in it.
Global buff size:	Global buffer size.
Data	Maximum data link packet size of any interface.
Header	Sum of the maximum hardware, MAC, and data link headers.
Wrap	Allowance given for MAC, LLC, or Network layer headers due to protocol wrapping.
Trailer	Sum of the largest MAC and hardware trailers.
Total	Overall size of each packet buffer

Network

Use the **network** command to enter the console environment for supported networks, such as X.25 networks. This command obtains the console prompt for the specified interface. From the prompt, you can display statistical information, such as the routing information fields for Token-Ring networks.

At the GWCON prompt (+), enter the **configuration** command to see the protocols and networks for which the router is configured. See "Configuration" on page 6-4 for more information on the configuration command.

Enter **interface** at the + prompt for a display of the networks for which the router is configured.

Enter the GWCON **network** command and the number of the interface you want to monitor. For example:

```
+network 3
X.25>
```

In the example, the X.25> prompt is displayed. You can then view information about the X.25 interface by entering the X.25 monitoring commands.

Exiting the Interface Console Process

To exit the interface console process and return to the OPCON process:

1. Return to the GWCON process by entering the **exit** command. For example:

   `>X.25>`**`exit`**

2. Return to the OPCON process by entering the OPCON intercept character.
   (**`Ctrl`**   **`P`** is the default intercept character.)

After identifying the interface number of the interface you want to monitor, for
interface-specific information, see the monitoring chapter in this manual for the
specified network or link-layer interface. Console support is offered for the
following network and link-layer interfaces:

- Ethernet
- Frame Relay
- PPP
- SDLC
- SDLC Relay (SRLY)
- Token Ring
- V.25 bis
- X.25
- ATM
- ISDN

Syntax: <u>net</u>work *interface#*

Example: **`network 2`**

   `ETH>`

Protocol

Use the **protocol** command to communicate with the router software that
implements the network protocols installed in your router. The **protocol** command
accesses a protocol's command environment. After you enter this command, the
prompt of the specified protocol appears. From the prompt, you can enter
commands that are specific to that protocol.

Enter the protocol number or short name as part of the command. To obtain the
protocol number or short name, enter the CONFIG command environment
(`Config>`), and then enter the **list configuration** command. See "Entering and
Exiting CONFIG" on page 3-6 for instructions on accessing `Config>`. To return to
GWCON, enter **exit**.

See the corresponding monitoring chapter in this manual or in the *Protocol
Configuration and Monitoring Reference* for information on a specific protocol's
console commands.

Syntax: <u>p</u>rotocol *prot#*

Example: **`protocol 7`**

   `IPX>`

The following table lists examples of protocol numbers and names.

Protocol Number	Protocol Short Name	Accesses the following protocol process
0	IP	IP (Internet Protocol)
3	ARP	ARP (Address Resolution Protocol)
4	DN	DNA Phase IV
6	VIN	Banyan VINES
7	IPX	IPX (Novell NetWare Internetwork Packet Exchange)
8	OSI	ISO CLNP/ESIS/ISIS
9	DVMRP	Distance Vector Multicast Routing Protocol
10	BGP	Border Gateway Protocol
11	SNMP	SNMP (Simple Network Management Protocol)
12	OSPF	OSPF (Open Shortest Path First)
20	SDLC	SDLC Relay
22	AP2	AppleTalk Phase 2
23	ASRT	Adaptive Source Routing Transparent Bridge
24	HST	TCP/IP Host Services
25	LNM	LAN Network Manager
26	DLSW	Data Link Switching
27	XTP	X.25 Transport Protocol
28	APPN HPR	APPN High Performance Routing
30	APPN ISR	APPN Intermediate Session Routing

Queue

Use the **queue** command to display statistics about the length of input and output queues on the specified interfaces. Information about input and output queues provided by the queue command includes:

- The total number of buffers allocated
- The low-level buffer value
- The number of buffers currently active on the interface.

To display information about one interface only, enter the interface or network number as part of the command. To obtain the interface number, use the GWCON **configuration** command.

Syntax: queue *interface#*

Example: **queue**

```
             Input Queue    Output Queue
Nt Interface Alloc Low Curr  Fair Curr
0  Eth/0        30  10   30    30    1
1  PPP/0        24   4   24     4    0
2  FR/0         24   4   24     5    0
```

Nt   Network interface number associated with the software.

Interface   Type of interface.

Input Queue:

Alloc	Number of buffers allocated to this device.
Low	Low water mark for flow control on this device.
Curr	Current number of buffers on this device. The value will be 0 if the device is disabled.
Output Queue:	
Fair	Fair level for the length of the output queue on this device.
Curr	Number of packets currently waiting to be transmitted on this device. For locally originated packets, the eligibility discard depends on the global low water mark described in the **memory** command.

The router attempts to keep at least the Low value packets available for receiving over an interface. If a packet is received and the value of Curr is less than Low, then the packet will be subject to flow control. If a buffer subject to flow control is to be queued on this device and the Curr level is greater than Fair, then the buffer is dropped instead of queued. The dropped buffer is displayed in the Output Discards column of the **error** command. It will also generate ELS event GW.036 or GW.057.

Due to the scheduling algorithms of the router, the dynamic numbers of Curr (particularly the Input Queue Curr) may not be fully representative of typical values during packet forwarding. The console code runs only when the input queues have been drained. Thus, Input Queue Curr will generally be nonzero only when those packets are waiting on slow transmit queues.

Statistics

Use the **statistics** command to display statistical information about the network software, such as the configuration of the networks in the router.

To display information about one interface only, enter the interface or network number as part of the command. To obtain the interface number, use the GWCON **configuration** command.

Syntax: statistics *interface#*

Example: `statistics`

```
              Unicast  Multicast     Bytes   Packets    Bytes
Nt Interface  Pkts Rcv  Pkts Rcv   Received    Trans    Trans
0  Eth/0          137         1       8832     1068    65297
1  PPP/0            0         0          0        0        0
2  PPP/1            0         0          0        0        0
```

Nt	Network interface number associated with the software.
Interface	Type of interface.
Unicast Pkts Rcv	Number of non-multicast, non-broadcast specifically-addressed packets at the MAC layer.
Multicast Pkts Rcv	Number of multicast or broadcast packets received.
Bytes Received	Number of bytes received at this interface at the MAC layer.

Packets Trans	Number of packets of unicast, multicast, or broadcast type transmitted.
Bytes Trans	Number of bytes transmitted at the MAC layer.

Test

Use the **test** command to verify the state of an interface or to enable an interface that was previously disabled with **disable** command. If the interface is enabled and passing traffic, the **test** command will remove the interface from the network and run self-diagnostic tests on the interface.

Enter the interface or network number as part of the command. To obtain the interface number, use the GWCON **configuration** command. When testing starts, the console displays the following message:

```
Testing net 0 TKR/0...
```

When testing completes, fails, or GWCON times out (after 30 seconds), the following possible messages are displayed:

```
Testing net 0 Eth/0 ...successful

Testing net 0 Eth/0 ...failed

Testing net 0 Eth/0 ...still testing

Network is already undergoing test, attempting restart
```

Some interfaces may take more than 30 seconds before testing is done.

Syntax:　　<u>test</u>　*interface#*

Example:　**test 0**

Note: For this command to work, you must enter the ***complete*** spelling of the command followed by the interface number.

Chapter 7. The MONITR Process

This chapter explains how to use the MONITR process and how to control the way
MONITR collects and displays messages. (Refer to Chapter 8, "Using and
Configuring the Event Logging System (ELS)" for information about ELS and
message formats. Refer also to the *IBM Nways Event Logging System Messages
Guide* for a description of each message.) This chapter includes the following
sections:

* "What is MONITR?"
* "Commands Affecting MONITR" on page 7-2
* "Entering and Exiting MONITR" on page 7-2
* "Receiving MONITR Messages" on page 7-2

What is MONITR?

The MONITR process provides a view of activity inside the router and the networks.
MONITR also displays logging messages from software that still uses the old
logging system. MONITR fits into the router software structure as shown in
Figure 7-1.
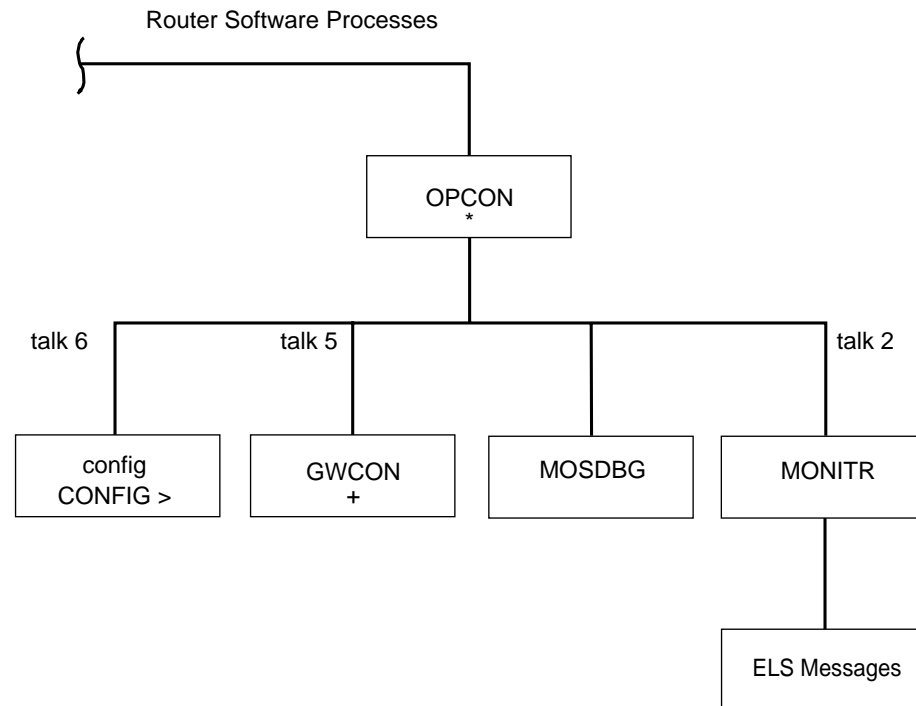
Figure 7-1. MONITR in the Router Software Structure

7-1

Commands Affecting MONITR

The following commands affect the MONITR process:

- OPCON commands:

    – **divert** temporarily diverts output to a different device.
    – **flush** causes MONITR to discard the messages it collects.
    – **halt** reverses the action of the divert command.
    – **talk** causes MONITR to display its output.

- CONFIG **set logging disposition** command sets the initial device to which MONITR sends its output.

Entering and Exiting MONITR

To enter the MONITR process from OPCON:

1. At the OPCON prompt, enter the status command to find the PID (process ID) of MONITR.

    ```
    * status
    ```

2. Enter the **talk** command and the PID number to enter the MONITR environment.

    ```
    * talk 2
    ```

MONITR does not display any prompt and you cannot enter any commands; however, the console begins to display the messages MONITR has accumulated.

To exit MONITR and return to OPCON, enter the OPCON intercept character (the default is **Ctrl** **P**).

Receiving MONITR Messages

To receive MONITR messages at your console, contact MONITR as described in the previous section. Then MONITR displays all the messages it has recorded since it was last invoked. While you are connected to MONITR, it displays all messages as they arrive.

You can use the OPCON **divert** and **halt** commands to view MONITR messages while you are doing something else with the router. Permitted devices divert output to TTY0 (the local console), TTY1, or TTY2 (the remote consoles).

Chapter 8. Using and Configuring the Event Logging System (ELS)

This chapter describes the Event Logging System (ELS) and its configuration. The ELS continually logs all events, filtering them according to parameters that you select. A combination of the GWCON counters and the ELS provides information for monitoring the health and activity of the system. The information is divided into the following sections:

- "What is ELS?"
- "Entering and Exiting the ELS Configuration Environment" on page 8-2
- "Event Logging Concepts" on page 8-3
- "ELS Configuration Commands" on page 8-7

What is ELS?

ELS is a monitoring system and an integral part of the router operating system. ELS manages the messages logged as a result of router activity. Using ELS commands, you can set up a configuration that sorts out only those messages that are important to you. You can display the messages on the console terminal screen or send the messages to a network management station using Simple Network Management Protocol (SNMP) traps.

The ELS system and the GWCON counters are the best troubleshooting tools you have to isolate problems in the router. A quick scan of the event messages will tell you whether or not the router has a problem and basically where to start looking for it.

In the ELS configuration environment, the commands are used to establish a default configuration. This default configuration does not take effect until you reinitialize the router.

Occasionally, it is necessary to temporarily view messages other than what was set up in the ELS configuration environment without having to reinitialize the router. The ELS console environment is used to:

- Temporarily change the default ELS display settings

  - Changes made in the ELS console environment take effect immediately

  - Changes made using console commands are not stored in non-volatile configuration storage.

- View statistical information regarding ELS uses of dynamic RAM

Note: Specific ELS messages are described in the *IBM Nways Event Logging System Messages Guide*.

ELS is a subprocess that you access from the OPCON process. ELS fits into the router software structure as shown in Figure 8-1 on page 8-2.

Router Software Processes
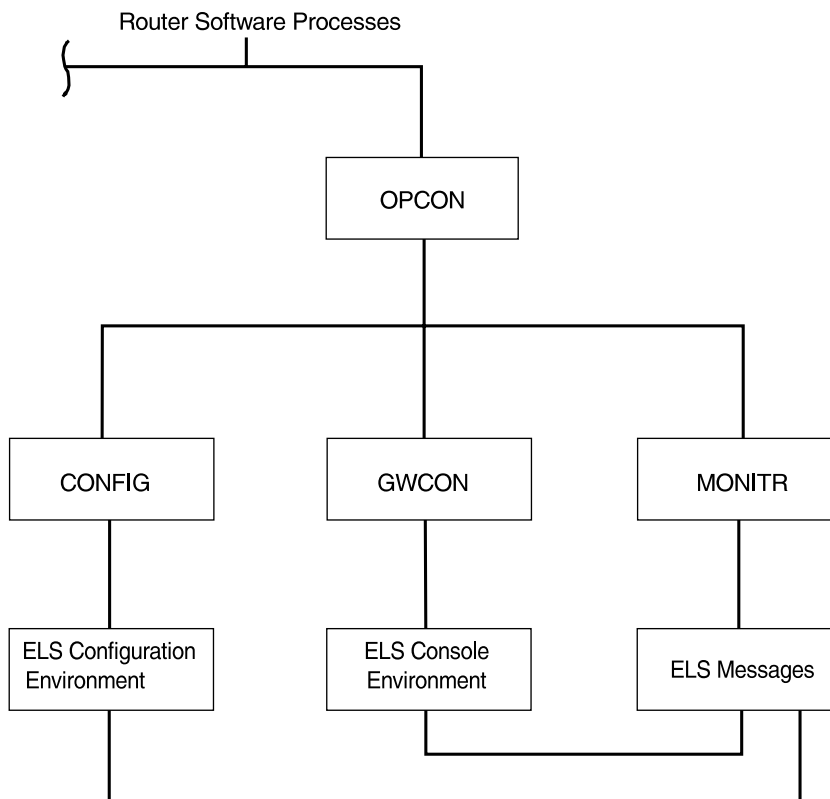
Figure 8-1. ELS in the Router Software Structure

Entering and Exiting the ELS Configuration Environment

The ELS configuration environment (available from the CONFIG process) is characterized by the `ELS Config>` prompt. Commands entered at this prompt create the ELS default state that takes effect after restarting the router. These commands are described in greater detail later in this chapter.

Configuration commands that have subsystem, group, or event as a parameter are executed in the following order:

- Subsystem
- Group
- Event

To set a basic ELS configuration, enter the **display subsystem all standard** command at the `ELS Config>` prompt. This command configures the ELS to display messages from all subsystems with the STANDARD logging level (that is, all errors and unusual informational comments).

Note: The router does not have a default ELS configuration. You must enter the ELS configuration environment and set the default state.

This section describes how to enter and exit the ELS configuration and console environments.

ELS Configuration Environment

To enter the ELS configuration environment from OPCON:

1. At the OPCON prompt, enter the **status** command to find the PID (process ID) of CONFIG.

   `* status`

2. Enter the **talk** command and the PID for CONFIG.

   `* talk 6`

   The console displays the CONFIG prompt (`Config>`). If the prompt does not appear when you first enter CONFIG, press **Return**.

3. At the CONFIG prompt, enter the following command to access ELS:

   `Config> eve`

   The console displays the ELS configuration prompt (`ELS config>`). Now, you can enter ELS configuration commands.

To leave the ELS configuration environment, enter the **exit** command. This command is described in this chapter.

Event Logging Concepts

This section describes how events are logged and how to interpret messages. Also described are the concepts of subsystem, event number, and logging level. A large part of ELS function is based on commands that take the subsystem, event number, and logging level as parameters.

Causes of Events

Events occur continuously while the router is operating. They can be caused by any of the following reasons:

- System activity
- Status changes
- Service requests
- Data transmission and reception
- Data and internal errors

When an event occurs, ELS receives data from the system that identifies the source and nature of the event. Then ELS generates a message that uses the data received as part of the message.

Interpreting a Message

This section describes how to interpret a message generated by ELS. Figure 8-2 shows the message contents.

```
        GW.019:        Slf tst nt 1 int ETH/0
          ↑   ↑                        ↑
Subsystem     |                   Message Text
          Event
         Number
```
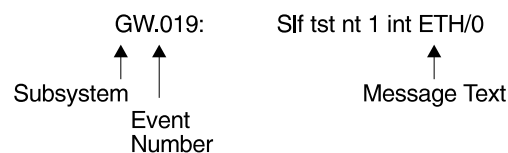
Figure 8-2. Message Generated by an Event

The information illustrated in Figure 8-2 as well as the ELS logging level information displayed with the **list subsystem** command is as follows:

Subsystem
Subsystem is a predefined short name for a router component, such as a protocol or interface. In Figure 8-2 on page 8-3, **GW** identifies the subsystem through which this event occurred.

Other examples of subsystems include IP, TKR, and X25. On a particular router, the actual subsystems present depend on the hardware and software configured for that router. You can use the **list subsystem** command described in this chapter to see a list of the subsystems on your router.

Enter the subsystem as a parameter to an ELS command when you want the command to affect the entire subsystem. For example, the ELS command **display subsystem GW** causes all events that occur through the GW subsystem to be displayed.

Event Number
Event Number is a predefined, unique, arbitrary number assigned to each message within a subsystem. In Figure 8-2 on page 8-3, **19** is the event number within the GW subsystem. You can see a list of all the events within a subsystem by using the **list subsystem** command, where *subsystem* is the short name for the subsystem.

The event number always appears with a subsystem, separated by a period. For example: **GW.019**. The subsystem and event number together identify an *individual* event. They are entered as a parameter to certain ELS commands. When you want a command to affect only the specified event, enter the subsystem and event number as a parameter for the ELS command.

Logging Level
Logging level is a predefined setting that classifies each message by the type of event that generated it. This setting is displayed whenever you use the **list subsystem** ELS console command. Table 8-1 on page 8-5 lists the logging levels and types.

Table 8-1. Logging Levels

Logging Level	Type
UI ERROR	Unusual internal errors
CI ERROR	Common internal errors
UE ERROR	Unusual external errors
CE ERROR	Common external errors
ERROR	Includes all error levels above
UINFO	Unusual informational comment
CINFO	Common informational comment
INFO	Includes all comment levels above
STANDARD	Includes all error levels and all informational comment levels (default)
PTRACE	Per packet trace
UTRACE	Unusual operation Trace message
CTRACE	Common operation Trace message
TRACE	Includes all trace levels above
ALL	Includes all logging levels

In Table 8-1, ERROR, INFO, TRACE, STANDARD, and ALL are aggregates of other logging level types. STANDARD is the recommended default.

The logging level setting affects the operation of the following commands:

- **Display subsystem**
- **Nodisplay subsystem**
- **Trap subsystem**
- **Notrap subsystem**

The logging level is set for a particular command when you specify it as a parameter to one of the above commands. For example:

`display subsystem` TKR ERROR

Including the logging level on the command line modifies the **display** command so that whenever an event with a logging level of either UI-ERROR or CI-ERROR occurs through subsystem TKR, the console displays the resulting message.

You cannot specify the logging level for operations affecting groups or events.

Message Text

Message Text appears in short form. In Figure 8-2 on page 8-3, `Slf tst nt 1 int ETH/0` is the message generated by this event. Variables, such as *source_address* or *network*, are replaced with actual data when the message displays on the console.

The variable *error_code* is referred to by some of the Event Logging System message descriptions (usually preceded by rsn or reason). They indicate the type of packet error detected. Table 8-2 on page 8-6 describes the error or packet completion codes. Packet completion codes indicate the disposition of the packets that arrive at the router.

Table 8-2. Packet Completion Codes (Error Codes)	
Code	**Meaning**
0	Packet successfully queued for output
1	Random, unidentified error
2	Packet not queued for output due to flow control reasons
3	Packet not queued because network is down
4	Packet not queued to avoid looping or bad broadcast
5	Packet not queued because destination host is down (only on networks where this can be detected)

ELS displays network information as follows:

`nt 1 int Eth/0` (or) `network 1, interface Eth/0,`

where:

- 1 is the network number (each network on the router is numbered sequentially from zero).

- 0 is the unit number (the interfaces of each hardware type are numbered sequentially from zero).

Ethernet and 802.5 hardware addresses appear as a long hexadecimal number.

IP (Internet Protocol) addresses are printed as 4 decimal bytes separated by periods, such as 18.123.0.16.

Groups

Groups are user-defined collections of events that are given a name, the group name. Like the subsystem, subsystem and event number, and logging level, you can use the group name as a parameter to ELS commands. However, there are no predefined group names. You must create a group before you can specify its name on the command line.

To create a group, use the **add** configuration command described in this chapter, specify the name you want to call the group, and then specify the events you want to be part of the group. The events you add to the group can be from different subsystems and have different logging levels.

After creating a group, you can use the group name to manipulate the events in the group as a whole. For example, to turn off display of all messages from events that have been added to a group named grouptwo, include the group name on the command line, as follows:

`nodisplay group grouptwo`

To delete a group, use the **delete** command.

ELS Configuration Commands

Table 8-3 summarizes the ELS configuration commands. The remainder of this section describes each one in detail. After accessing the ELS configuration environment, you can enter ELS Configuration commands at the `ELS Config>` prompt.

Table   8-3. ELS Configuration Command Summary

Command	Function
? (Help)	Lists the ELS configuration commands or lists the options associated with specific commands.
Add	Adds an event to an existing group or creates a new group.
Clear	Clears all ELS configuration information.
Default	Resets the display or trap setting of an event, group, or subsystem.
Delete	Deletes an event number from an existing group or deletes an entire group.
Display	Enables message display on the console monitor.
Exit	Exits the ELS configuration process and returns you to the CONFIG process.
List	Lists information on ELS settings and messages.
Nodisplay	Disables message display on the console.
Notrace	Controls disablement of packet trace events.
Notrap	Keeps messages from being sent out in SNMP traps.
Set	Sets the pin parameter, the timestamp feature, and atm packet tracing options.
Trace	Controls enablement of packet trace events.
Trap	Allows messages to be sent to a network management workstation in SNMP traps.
View	Allows viewing of traced packets.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
CLEAR
DEFAULT
DELETE
DISPLAY
LIST
NODISPLAY
NOTRACE
NOTRAP
SET
TRACE
TRAP
VIEW
EXIT
```

Example: `list ?`

```
ALL
GROUPS
PIN
STATUS
SUBSYSTEM
```

Add

Use the **add** command to add an individual event to an existing group or to create a new group. Group names must start with a letter and are case sensitive. You cannot append an entire subsystem to a group.

Syntax: a̲dd   *group_name subsystem.event_number*

Example: `add MyGroup gw.019`

> **Note:** If the specified group does not exist, the following prompt asks you to confirm the creation of a new group:
>
> `Group not found. Create new group? (yes or no)`

Clear

Use the **clear** command to clear all of the ELS configuration information.

Syntax: c̲lear

Example: `clear`

```
You are about to clear all ELS configuration information
Are you sure you want to do this (Yes or No):
```

Default

Resets the display or trap setting of an event, group, or subsystem back to a disabled state.

Syntax: d̲ef̲ault   d̲isplay
                        t̲rap

display   *event* OR *group* OR *subsystem*
   Controls the output of the display of messages to the console.

   Example: `default display event snmp.016`

trap   *event* OR *group* OR *subsystem*
   Controls the generation of traps to the network management station.

   Example: `default trap subsystem ip`

Delete

Use the **delete** command to delete an event number from an existing group or to delete the entire group. If the specified event is the last event to be deleted in a group, you will be notified. If *all* is specified instead of *subsystem.event_number*, a prompt asks you to confirm the deletion of the entire group.

Syntax: d̲el̲ete   *group_name subsystem.event_number*

Example: `delete groupa gw.019`

Display

Use the **display** command to enable message displaying on the console monitor for specific events, groups, or subsystems.

Syntax: <u>di</u>splay   event . . .
                     group . . .
                     subsystem . . .

event *subsystem.event#*
    Displays messages of the specified event (*subsystem.event#*).

    Example: `display event gw.019`

group *groupname*
    Displays messages of a specified group (*groupname*).

    Example: `display group groupb`

subsystem *subsystemname*
    Displays messages associated with the specified subsystem. The following is a list of subsystems that are supported on the router. To find out which subsystems are on your router, type **list subsystems**.

    Note: Although ELS supports all of these subsystems, not all devices support all subsystems See *ELS Messages* for the most current list of supported subsystems.

Subsystem	**Description**
AI	Auto-device Install
All	All subsystems

> **Note:** Do not display all subsystems for extended periods of time when the router is forwarding live protocol traffic because this causes the router to spend an excessive amount of time communicating with the console. Never display all subsystems when you are communicating with the router via a remote console. This causes the router to spend most of its time communicating with the remote console.

AP2	AppleTalk Phase 2
ARP	Address Resolution Protocol
APPN	Advanced Peer-to-Peer Networking
ATM	Asynchronous Transfer Mode
BAN	Boundary Access Node
BGP	Border Gateway Protocol
BR	Bridging/Routing
BRS	Bandwidth Reservation
BTP	BOOTP relay agent
CLNP	ISO 8473 - CLNP
COMP	Data Compression
DLS	Data Link Switching
DN	DECnet
DNAV	DNA Phase V
DVM	DVMRP Multicast Routing Protocol
ESIS	ISO 9542 - ESIS Protocol
ETH	Ethernet handler
EZ	EasyStart
FLT	Filter library

FRL	Frame Relay
GW	Router base and network library
ICMP	Internet Control Message Protocol
ILMI	Interim Local Management Interface
IP	Internet Protocol
IPPN	IP Protocol Net
IPX	Internetwork Packet Exchange Protocol
ISDN	Integrated-services Digital Network
ISIS	ISO 10589 - ISIS Protocol
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LES	LAN Emulation Server
LLC	Logical Link Control
LNM	LAN Network Manager
MCF	MAC Filtering
MSPF	OSPF Multicast extensions
NBS	NetBIOS Support Subsystem
NOT	Non-supported Protocol Forwarder
OSPF	Open SPF-based Routing Protocol
PPP	Point-to-Point Protocol
RIP	IP Routing Information Protocol
R2MP	AppleTalk Phase 2 Routing Table Management Protocol
SAAL	Signaling ATM Adaptation Layer
SDLC	IBM SDLC
SL	Serial Line Handler
SNMP	Simple Network Management Protocol
SRLY	SDLC Relay
SRT	Source Routing Transparent Bridge
STP	Spanning Tree Protocol
SVC	Switched Virtual Connection
TCP	Transport Control Protocol
TFTP	Trivial File Transfer Protocol
TKR	Token Ring Handler
UDP	User Datagram Protocol
VIN	Banyan VINES
V25B	CCITT/ITU V.25 bis
WRS	WAN Restoral/Reroute
XN	XNS/IPX/DDS common processing
XNS	Xerox Networking Systems Protocol
X25	X.25 Protocols
X251	X.25 Physical Layer
X252	X.25 Frame Layer
X253	X.25 Packet Layer
XTP	X.25 Transport Protocol
ZIP2	AppleTalk Phase 2 Zone Information Protocol
ILMI	ATM Interim Local Management Interface
LEC	ATM LAN Emulation Client

Example: `display subsystem tkr`

List

Use the **list** command to get updated information regarding ELS settings and listings of selected messages.

Syntax: list   all
                        groups
                        pin
                        status
                        subsystem
                        subsystem . . .
                        subsystems all
                        trace-status

all

    Lists information from all the **list** categories.

    Example: `list all`

groups

    Lists the user-defined group names and contents.

    Example: `list groups`

```
Group: test
GW.019
```

pin

    Lists the current number of ELS event messages sent in SNMP traps (per second).

    Example: `list pin`

```
Pin: 100 events/second
```

status

    Lists the subsystems, groups, and events that have been modified by the **display**, **nodisplay**, **trap**, and **notrap** commands.

    Example: `list status`

subsystem

    Lists names, events, and descriptions of all subsystems.

    Example: `list subsystem`

                (Example output from a **list subsystem** command can be found beginning on page 9-9.)

subsystem *subsystem*

    Lists all events in a specified subsystem.

    Example: `list subsystem gw`

```
Event     Level     Message

GW.001    ALWAYS    Copyright 1984 Mass Institute of Technology
GW.002    ALWAYS    Portable CGW %s Rel %s strtd
GW.003    ALWAYS    Unus pkt len %d nt %d int %s/%d
GW.004    ALWAYS    Sys %s q adv alloc %d excd %d
GW.005    ALWAYS    Bffrs: %d avail %d idle   fair %d low %d
GW.006    C-INFO    Pkt frm nt %d int %s/%d for uninit prt, disc
GW.007    C-INFO    Ip err %x nt %d int %s/%d
GW.008    U-INFO    Ip ovfl nt %d int %s/%d, disc
GW.009    UI-ERROR  Nt dwn ip rstrt nt %d int %s/%d
GW.010    UI-ERROR  Ip q len %d no ip buf nt %d int %s/%d
GW.011    U-INFO    Op err %x hst %wo nt %d int %s/%d
GW.012    U-INFO    Op err cnt excd hst %wo nt %d int %s/%d
GW.013    U-INFO    Rtrns cnt excd hst %wo nt %d int %s/%d
GW.014    UI-ERROR  Nt dwn op rstrt nt %d int %s/%d
GW.015    UI-ERROR  Nt dwn to hst %wo nt %d int %s/%d
GW.016    U-INFO    Op ovfl to hst %wo nt %d int %s/%d
GW.017    UE-ERROR  Intfc hdw mssng nt %d int %s/%d
GW.018    U-TRACE   Strt nt slf tst nt %d int %s/%d
GW.019    C-INFO    Slf tst nt %d int %s/%d
GW.020    U-TRACE   Nt pss slf tst nt %d int %s/%d
GW.021    UE-ERROR  Nt up nt %d int %s/%d
GW.022    U-TRACE   Nt fld slf tst nt %d int %s/%d
```

subsystems <u>all</u>
> Lists all events in all subsystems.

> **Example:**   `list subsystems all`

trace-status
> Displays information on the status of packet tracing, including configuration and run-time information.

> **Example:**   `list trace-status`

```
------------------------ Configuration ----------------------------
Trace Status:ON  Wrap Mode:ON  Decode Packets:ON  HD Shadowing:ON
RAM Trace Buffer Size:100000  Maximum Trace Buffer File Size:10000000
Max Packet Bytes Trace:256  Default Packet Bytes Traced:100
Trace File Record Size:2048  Stop Trace Event: TCP.013
Maximum Hours to HD Shadow:  1
```

Nodisplay

Use the **nodisplay** command to select and turn off messages displaying on the console monitor.

Syntax: <u>nod</u>isplay   <u>event</u>. . .
                            <u>group</u> . . .
                            <u>subsystem</u> . . .

event *subsystem.event#*
> Suppresses the displaying of a specified event (*subsystem.event#*).

> **Example:**   `nodisplay event gw.019`

group *groupname*
> Suppresses the displaying of messages that were previously added to the specified group (*groupname*).

> **Example:**   `nodisplay group groupb`

subsystem *subsystemname*
> Suppresses the displaying of messages associated with the specified subsystem.

> **Example:**   `nodisplay subsystem tkr`

Notrace

Disables packet trace for the specified event/subsystem/group.

Syntax: <u>not</u>race   event. . .

                    group. . .

                    subsystem. . .

event subsystem.*event#*
>   Suppresses the sending of packet trace data for the specified event#

>   **Example: `notrace event atm.088`**

group *groupname*
>   Suppresses the sending of packet trace data that was previously added to the
>   specified group (groupname).

>   **Example: `notrace group groupb`**

subsystem *subsystemname*
>   Suppresses the sending of packet trace data for the specified subsystem
>   (subsystemname).

>   **Example: `notrace subsystem atm`**

Notrap

Use the **notrap** command to select and turn off messages so that they are no
longer sent to a network management workstation in SNMP traps.

Syntax: <u>not</u>rap   <u>event</u>. . .
                    <u>group</u> . . .
                    <u>subsystem</u> . . .

event *subsystem.event#*
>   Suppresses the sending of the specified message in an SNMP trap
>   (*subsystem.event#*).

>   **Example: `notrap event gw.019`**

group *groupname*
>   Suppresses the sending of messages in SNMP traps that were previously
>   added to the specified group (*groupname*).

>   **Example: `notrap group groupb`**

subsystem *subsystemname*
>   Suppresses the sending of messages in SNMP traps that are associated with
>   the specified subsystem.

>   **Example: `notrap subsystem tkr error`**

Set

Use the **set** command to set the maximum number of traps per second, to set the
timestamp feature, or to set tracing options for ATM devices.

Syntax: <u>s</u>et   pin . . .
                    timestamp . . .
                    trace . . .

pin *max_traps*

Use the **set pin** command to set the pin parameter to the maximum number of traps that can be sent on a per-second basis. Internally, the pin resets every tenth of a second. (One tenth of the number (*max_traps*) is sent every tenth of a second.)

Example: `set pin 100`

timestamp   timeofday OR uptime OR off

Allows you to turn on message timestamping so that either the time of day or uptime (number of hours, minutes, and seconds, but no date, since the router was last initialized) appears next to each message. Set timestamp can also be turned off.

Use the **set timestamp** command to enable one of the following timestamp options.

Example: `set timestamp  timeofday`

timeofday

Adds an HH:MM:SS prefix to each ELS message indicating the time of the occurrence during a 24-hour day.

uptime

Adds an HH:MM:SS prefix to each ELS message indicating the time of the occurrence during a 100-hour cycle. After 100 hours of uptime, the uptime counter returns to zero to begin another 100-hour cycle.

off

Turns off the ELS timestamp prefix.

trace

Use the **set trace** command to configure tracing options for ATM devices. When tracing options are configured from the monitoring console, the changes take effect immediately, and return to their previously configured settings when the device is rebooted.

Syntax: <u>s</u>et trace   decode
default-bytes-per-port
disk-shadowing
max-bytes-per-port
memory-trace-buffer-size
off
on
reset
stop-event
wrap-mode

decode *off/on*

Turns packet decoding on or off. Packet decoding is not supported by all components.

default-bytes-per-pkt *bytes*

Sets the default number of bytes traced. This value is used if a value is not specified by the component doing the tracing.

disk-shadowing *off / on* OR *record-size* OR *time-limit*

Turns disk shadowing on or off, sets the maximum trace file size, or sets the maximum time for disk-shadowing traces.

off / on   Turns disk shadowing on or off. If disk shadowing is enabled, trace records are copied to the hard disk. Once a traced record is copied to the hard disk, it can no longer be viewed from the console.

> **Note:** Disk shadowing should be set to OFF whenever the WRITE, TFTP software, RETRIEVE system dump, or COPY software commands are issued.

> **Example: `set trace off`**

record-size Sets the record size for trace file records:

> **Valid Values**   1024, 2048, or 4096 bytes
> **Default**   2048 bytes

> **Notes:**

> 1. If a trace file already exists, "Cannot change Record Size without first deleting the existing Trace File" is displayed and record size is not changed.

> 2. If you configure a record size and a trace file already exists, the trace will use the record size of the existing file.

> **Example: `set trace record-size 4096`**

time-limit   Sets the maximum time for disk-shadowing of traces:

> **Valid Values**   1 - 72 hours
> **Default**   24 hours

> **Note:** Disk shadowing stops (tracing continues) after this time has elapsed. The actual time is reset to 0 when disk shadowing is turned on again.

> **Example: `set trace disk-shadowing time 36`**

max-bytes-per-pkt *bytes*
    Sets the maximum number of bytes traced for each packet.

memory-trace-buffer-size *bytes*
    Sets the size, in bytes, of the RAM trace buffer.

    off   Disables packet tracing.
    on   Enables packet tracing.
    reset   Clears the trace buffer and resets all associated counters.

stop-event *event id*
    Stops tracing when an event (event id) occurs. Enter either an ELS event id (for example: TCP.013) or "None."   "None" is the default. Tracing stops only if the display of the particular ELS event is enabled.

    When a stop-event occurs, an entry is written to the trace buffer. The **view** command for this trace entry will display "Tracing stopped due to ELS Event Id: TCP.013."

> After tracing stops due to a stop-event, you must re-enable tracing with the **set trace on** command. (A restart will also re-enable tracing if enabled from the ELS Config> prompt.)
>
> **Example: `set trace stop-event TCP.013`**

wrap-mode *off/on*
> Turns the trace buffer wrap mode on or off. If wrap mode is on and the trace buffer is full, previous trace records will be overwritten by new trace records as necessary to continue tracing.

Example 1:   `set trace decode on`

Example 2:   `set trace default-bytes-per-packet 64`

Example 3:   `set trace off`

Trace

Enables packet trace for the specified event/subsystem/group. When the **trace** command is used from the ELS Config> prompt, the changes become part of the configuration, and a reboot is required to activate the changes.

See "Trace" on page 9-17 for a complete description of the **trace** command.

Trap

Use the **trap** command to select the message to be sent to the remote SNMP network management workstation. A remote SNMP network management workstation is an IP host in the network acting as an SNMP manager.

Syntax: trap   <u>event</u> . . .
                           <u>group</u> . . .
                           <u>subsystem</u> . . .

event *subsystem.event#*
> Causes the specified message (*subsystem.event#*) to be sent to a network management workstation in an SNMP trap.

Example: `trap event gw.019`

group *groupname*
> Allows messages that were previously added to the specified group to be sent to a network management workstation in an SNMP trap.

Example: `trap group groupb`

subsystem *subsystemname*
> Allows messages associated with the specified subsystem to be sent to a management station in an SNMP trap.

Example: `trap subsystem gw`

Note: Messages for the IP, ICMP, ARP and UDP subsystems cannot be sent in SNMP traps because these areas are or may be used in the process of sending the SNMP trap. This could lead to an infinite loop of traffic putting an undue strain on the router.

Exit

Use the **exit** command to return to the CONFIG prompt.

Syntax: <u>e</u>xit

Example: **exit**

Chapter 9. Monitoring the Event Logging System (ELS)

This chapter describes how to monitor events logged by ELS and how to use the ELS console commands. The information includes the following sections:

- "Using ELS"
- "Using ELS to Troubleshoot a Problem" on page 9-3
- "Entering and Exiting the ELS Console Environment" on page 9-5
- "ELS Console Commands" on page 9-6

If you need more information on the Event Logging System and how to interpret ELS event messages, refer to Chapter 8, "Using and Configuring the Event Logging System (ELS)."

Using ELS

To effectively use ELS, it is recommended you do the following:

- Know what you want to see before using the ELS system. Clearly define the problem or events that you want to see before using the MONITR process.

- Execute the command **nodisplay subsystem all all** to turn off all ELS messages.

- Turn on only those messages that relate to the problem you are experiencing.

- Use the *IBM Nways Event Logging System Messages Guide* to determine which messages you are seeing are normal.

When you initially view ELS from the MONITR process, you will see a considerable amount of information. Because the router cannot buffer and display every packet under moderate to heavy loads the buffers are flushed. When this occurs the following message is displayed:

```
xx messages flushed
```

The router does not save these messages so that they can be reviewed later. When this message appears, you may want to tailor the ELS output to display only that information that is important to the current task you are monitoring.

Managing ELS Message Rotation

It is also important to note that the ELS messages continually rotate through the router's buffers. To stop and restart the displaying of ELS messages, use the following key combinations:

Ctrl **S**	to pause scrolling
Ctrl **Q**	to resume scrolling
Ctrl **P**	to go back to the last process

You may also want to capture the ELS output to a file. You can do this by starting a script file or log file from your location when Telneting to a router. You can also do this by attaching a PC to the router's console port and starting a log file from within the terminal emulation package. This information is needed to help Customer Service diagnose a problem.

Capturing ELS Output Using a Telnet Connection on a UNIX Host

You can use a Telnet connection on an AIX or UNIX host to capture the ELS messages on your screen to a file on the host. Before beginning, make sure you have set up ELS for the messages you want to capture by using the ELS console commands in this chapter.

To capture the ELS output to a file on an AIX or UNIX host, follow these steps:

1. From the host, enter **telnet** *router_ip_addr* **| tee** *local_file_name*

   router_ip_addr is the IP address of the router

   local_file_name is the name of the file on the host where you want the ELS messages to be saved.

   The **tee** command displays the ELS messages on your screen and, at the same time, copies them to the local file.

2. From the OPCON prompt (*), enter **t 2**. This accesses the MONITR process, which is the process that displays ELS messages on your screen. Depending on which ELS messages you configured, you should see ELS messages appearing on the screen.

   As long as you are in the MONITR process, all ELS messages will be written to the local file. When you exit the MONITR process (by entering `Ctrl` `P`) or terminate the Telnet session, the logging of messages to the local file will stop.

Configuring ELS So Event Messages Are Sent In SNMP Traps

ELS can be configured so that event messages are sent to a network management workstation in an SNMP enterprise-specific trap. These traps are useful for reporting status and diagnostic results, and are often used for remote monitoring of a 2216. When ELS is configured appropriately, an SNMP trap will be generated each time the selected event occurs. For more information about SNMP, see *Protocol Configuration and Monitoring Reference*.

To tell ELS that a specific event should be activated to be sent as an SNMP trap, at the ELS config> prompt or at the ELS> prompt, using IP as an example, type:

```
trap event ip.007
```

Note: If you are at the ELS config> prompt, you will need to reboot.

To enable the ELS enterprise-specific trap, follow these steps:

1. At the SNMP config> prompt, using **public** as an example, type:

   SNMP config> **add address public <network manager IP address>**

   SNMP config> **enable trap enterprise public**

   SNMP config> **set community access read_trap public**

   Note: You will need to reboot to activate these changes.

2. Enable your network management station to receive and properly display the enterprise-specific traps.

You can follow the steps above for trapping groups, subsystems, and events.

Using ELS to Troubleshoot a Problem

Events occur continuously while the router is operating. They can be caused by any of the following reasons:

- System activity
- Status changes
- Service requests
- Data transmission and reception
- Data and internal errors

When an event occurs, ELS receives data from the system that identifies the source and nature of the event. Then ELS generates a message that uses the data received as part of the message.

When trying to troubleshoot a particular problem, display those messages that relate to the problem. For example, if you are experiencing a problem with bridging, turn on the bridging messages:

> **display subsystem srt all**
> **display subsystem br all**

Initially, because of the rapid pace of messages scrolling across the screen, you may want to record the numbers you see and look those up in the manual. Once you become familiar with different types of messages being displayed for a particular protocol, you can turn on and turn off only those messages that contain the information that you require to troubleshoot a problem. The following sections list specific ELS examples. Keep in mind that different problems may require different steps.

ELS Example 1

You are interested in looking at the frequency of polling on a Token-Ring interface, and finding out whether the polls are successful.

```
ELS> nodisplay subsystem all all
```

```
ELS> display subsystem tkr all
```

Ctrl   **P**

∗ **t 2**

As the messages begin to scroll by, look for ELS message tkr.031.

ELS Example 2

SRTB bridging is not working.

1. Check the configuration.

2. Use the GWCON bridging console to verify that the bridging interfaces are enabled.

3. Enter:

```
* t 6

config> event

ELS config> nodisplay subsystem all all

ELS config> display subsystem srt all

ELS config> exit

config> Ctrl   P
```

4. Restart the routing subsystem. When the subsystem has restarted, enter the following:

```
* t 2
```

As the messages begin to scroll by, look for messages srt.071 through srt.075. If you see one of these messages, you are not licensed to use one or more of the bridging features.

ELS Example 3

Router cannot communicate with an IPX server on an Ethernet.

1. At the OPCON prompt, enter the **status** command to find the PID (process ID) of GWCON. (See step 1 on page 8-3 in ELS Configuration Environment for a sample output of the **status** command.)

```
* status
```

2. Enter the **talk** command and the PID for GWCON.

```
* talk 5
```

The console displays the GWCON prompt (+). If the prompt does not appear when you first enter GWCON, press **Return**.

3. At the GWCON prompt (+), enter **IPX** to access the IPX console prompt (IPX>).

4. At the IPX console prompt, enter the **slist** command to verify that the server is listed. (See the section on monitoring IPX in the *Protocol Configuration and Monitoring Reference* for information on the slist command.)

5. Check the IPX configuration.

6. Enter the following:

```
* t 5

+ event

ELS> nodisplay subsystem all all

ELS> display subsystem IPX all

ELS> display subsystem eth all

ELS> Ctrl   P

* t 2
```

As the messages begin to scroll by, look for ELS message eth.006. This indicates that the server has a bad econfig.

Entering and Exiting the ELS Console Environment

The ELS console environment (available from the GWCON process) is characterized by the ELS> prompt. Commands entered at this prompt modify the current ELS parameter settings. These commands are described Chapter 9, "Monitoring the Event Logging System (ELS)."

To enter the ELS console environment from OPCON:

1. At the OPCON prompt, enter the **status** command to find the PID (process ID) of GWCON. (See step 1 on page 8-3 in ELS Configuration Environment for a sample output of the **status** command.)

   `* status`

2. Enter the **talk** command and the PID for GWCON.

   `* talk 5`

   The console displays the GWCON prompt (+). If the prompt does not appear when you first enter GWCON, press **Return**.

3. At the GWCON prompt, enter the following command to access ELS:

   `+ event`

   The console displays the ELS console prompt (ELS>). Now, you can enter ELS console commands.

To leave the ELS console environment, enter the **exit** command.

ELS Console Commands

This section summarizes and then explains all the ELS console commands. After accessing the ELS Console environment, you can enter ELS console commands at the ELS> prompt.

Command	Function
? (Help)	Lists the ELS console commands or lists the options associated with specific commands.
Clear	Clears messages associated with specific events, groups, or subsystems.
Display	Enables message display on the console.
Exit	Exits the ELS console process and returns the user to GWCON.
List	Lists information on ELS settings and messages.
Nodisplay	Disables message display on the console.
Notrace	Disables trace event display on the console.
Notrap	Keeps messages from being sent out in SNMP traps to the network management workstation.
Remove	Frees up memory by erasing stored information.
Restore	Clears current settings and reloads initial ELS configuration.
Retrieve	Reloads the saved ELS configuration.
Save	Stores the current configuration.
Set	Sets the pin parameter and the timestamp feature. *
Statistics	Displays available subsystems and pertinent statistics.
Trace	Enables trace event display on the console.
Trap	Allows messages to be sent to a network management workstation in SNMP traps.
View	Allows viewing of traced packets.

Table   9-1. ELS Console Command Summary

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

```
                             CLEAR
                             DISPLAY
                             LIST
                             NODISPLAY
                             NOTRACE
                             NOTRAP
                             REMOVE saved state
                             RESTORE initial state
                             RETRIEVE saved state
                             SAVE current state
                             SET
                             STATISTICS
                             TRACE
                             TRAP
                             VIEW
                             EXIT
```

Example: list ?

```
                             ALL
                             ACTIVE
                             EVENT
                             GROUPS
                             PIN
                             SUBSYSTEMS
                             TRACE
```

Clear

Use the clear command to disable both the display and trap commands as they relate to specific events, groups, or subsystems.

Syntax: <u>c</u>lear   <u>event</u> . . .
                    <u>group</u> . . .
                    <u>subsystem</u> . . .

event *subsystem. event#*
    Disables the displaying or trapping of messages for the specified event (*subsystem.event#*).

    Example: clear event gw.019

group *group.name*
    Disables the displaying or trapping of messages for the specified group (*group.name*). specified group (*groupname*).

    Example: clear group groupb

subsystem *subsystem.name*
    Disables the displaying or trapping of messages associated with the specified subsystem (*logging level*). If you do not specify a logging level, all messages for that subsystem are disabled.

    Example: clear subsystem gw

Display

Use the display command to enable the message display on the console monitor for specific events.

Syntax: <u>d</u>isplay   <u>event</u> . . .
                      <u>group</u> . . .
                      <u>subsystem</u> . . .

event *subsystem.event#*
    Displays messages for the specified event (subsystem.event#).

Example: `display event gw.019`

group *group.name*
Displays the messages that were previously added to the specified group
(*group.name*).

Example: `display group groupb`

subsystem *subsystem.name*
Displays any messages associated with the specified subsystem (*logging
level*). If you do not specify a logging level, all messages for that subsystem
are turned on.

Example: `display subsystem tkr`

Files Trace TFTP

Use the **files trace tftp** command to retrieve trace files from the subdirectory
associated with:

- The currently active bank (bank A or bank B on the hard disk)
- Bank A on the hard disk
- Bank B on the hard disk
- The trace file stored in the Network Subdirectory (if there is no active bank)

Syntax: f̲iles t̲race t̲ftp   a̲ctive-bank ...
                                    bank-a̲ ...
                                    bank-b̲ ...
                                    n̲et-subdir ...

You are prompted for the *remote server IP address* and the *remote path/file name*.

active-bank   Retrieves the traces file from the currently active bank

bank-a   Retrieves the trace file from bank A.

bank-b   Retrieves the trace file from bank B.

net-subdir   Retrieves the trace file stored in the Network Subdirectory (if there
                  is no active bank).

Example: `files trace tftp active-bank 9.67.240.163 trace1.log`

List

Use the list command to get updated information regarding ELS settings and to get
listings of selected messages.

Syntax: l̲ist   a̲ll
                          a̲ctive . . .
                          e̲vent . . .
                          g̲roups . . .
                          p̲in
                          s̲ubsystems . . .
                          t̲race-status

all
Lists all subsystems, defined groups, enabled subsystems, enabled events,
and pins.

Example: `list all`

active *subsystem.name*
Displays the events that are active for a specific subsystem and the count of
the occurrence of the messages.

Example: `list active ip`

```
EventActiveCount
IP.00789354
ETH.009D10
Subsystem X25: no event active
```

event *subsystem.event#*
Displays the logging level, the message, and the count of the specified event.

Example: `list event ip.007`

```
Level: p-TRACE
Message: %I -> %I
Active:  Count: 84182
```

groups *group.name*
Displays the user-defined group names.

Example: `list groups`

pin

Lists the current number of ELS event messages sent per second in SNMP
traps. This is a threshold value that can be used to reduce the amount of
SNMP trap traffic.

Example: `list pin`

```
Pin: 100 events/second
```

subsystem *subsystem.name*
Lists event names, the total number of events that have occurred, and their
descriptions.

Note: The following example represents a list subsystem command and may
not reflect all of the subsystems and their event numbers. For the most recent
list of subsystems and event numbers, see *ELS Messages.*

Example: `list subsystem`

```
Name   Events  Description

ALL            All subsystems
APPN   28      Advanced Peer-to-Peer Networking
GW     78      Router base and network library
FLT    7       Filter Library
BRS    4       Bandwidth  Reservation
ARP    19      Address Resolution Protocol
IP     86      Internet Protocol
ICMP   20      Internet Control Message Protocol
TCP    55      TCP
UDP    4       User Datagram Protocol
BTP    5       BOOTP relay agent
RIP    19      IP Routing Information Protocol
OSPF   61      Open SPF-Based Routing Protocol
MSPF   15      OSPF Multicast extensions
TFTP   24      TFTP Protocol
SNMP   20      Simple Network Management Protocol
DVM    21      DVMRP Multicast Routing Protocol
IPX    89      Internetwork Packet Exchange Protocol
AP2    68      AppleTalk Phase 2
ZIP2   51      AppleTalk Phase 2 Zone Information Protocol
R2MP   38      AppleTalk Phase 2 Routing Table Management Protocol
VIN    79      Banyan VINES
SRT    87      Source Routing Transparent Bridge
```

```
STP    32     Spanning Tree Protocol
BR     29     Bridge/Routing
SRLY   28     SDLC Relay
ETH    47     Ethernet Handler
TKR    45     Token Ring Handler
X25    53     X.25 Protocols
ILMI    6     Interim Local Management Interface
SDLC   92     IBM SDLC
FRL    52     Frame Relay
PPP    93     Point-to-Point
X251   16     X.25-Physical-Layer
X252   34     X.25-Frame-Layer
X253   39     X.25-Packet-Layer
XTP    57     X.25 Transport Protocol
IPPN    4     IP Protocol Net
WRS    22     WAN Restoral/Reroute
LLC    33     Logical Link Control
MCF     9     MAC Filtering
SAAL   18     Signaling ATM Adaptation Layer
DLS   337     Data Link Switching
V25B   28     CCITT/ITU V.25bis
BAN    29     Boundary Access Node
ATM    13     Asynchronous Transfer Mode
LEC     7     ATM LAN Emulation Client
SVC     5     ATM Signalling
```

subsystem *subsystem.name*

Lists all events, logging levels, and messages for the specified subsystem.

Example: `list subsystem eth`

```
Event      Level      Message
ETH.001    P-TRACE    brd rcv unkwn type %X %E -> %E nt %d int %s/%d
ETH.002    UE-ERROR   rcv unkwn typ %X %E -> %E nt %d int %s/%d
ETH.010    C-INFO     LLC unk SAP %X %E -> %E nt %d int %s/%d
```

subsystems a̲ll

Lists all events, logging levels, and messages for every event that has occurred on the router.

Example: `list subsystems all`

trace-status

Displays information on the status of ATM packet tracing, including configuration and run-time information.

Example: `list trace-status`

```
------------------------ Configuration ----------------------------
Trace Status:ON  Wrap Mode:ON  Decode Packets:ON  HD Shadowing:ON
RAM Trace Buffer Size:100000  Maximum Trace Buffer File Size:10000000
Max Packet Bytes Trace:256  Default Packet Bytes Traced:100
Trace File Record Size:2048  Stop Trace Event: TCP.013
Maximum Hours to HD Shadow:  1
----------------------- Run-time Status ---------------------------
Packets in RAM Trace Buffer:1    Free Trace Buffer Memory:99958
Trace Errors:0  First Packet:1  Last Packet:1
Trace Records Stored on HD:8  Trace Buffer File Size:16560
HD-Shadowing Time Exceeded? NO Elapsed Time:  0 hr,  0 min, 10 sec
Has Stop Trace Event Occurred? NO
```

- "Trace Status" in the LIST TRACE-STATUS display will indicate OFF when STOP-ON-EVENT action occurs.

- "HD Shadowing" in the LIST TRACE-STATUS display will indicate OFF when STOP-ON-EVENT action occurs or when Time Limit is exceeded.

- "Trace Buffer File Size" will display "<wrapped>" when a wraparound has occurred in the trace file.

- If disk-shadowing time limit is exceeded, but there has not been a trace record written since the time expired, then "HD-Shadowing Time Exceeded? NO <Next trace will turn it OFF>" will be displayed. When the next trace record has been written, then "HD-Shadowing Time Exceeded? YES" will be displayed.

"ELS Config>LIST TRACE" command under Talk-6 displays information similar to the following:

```
------------------------ Configuration ----------------------------
Trace Status:ON  Wrap Mode:ON  Decode Packets:ON  HD Shadowing:ON
RAM Trace Buffer Size:100000  Maximum Trace Buffer File Size:10000000
Max Packet Bytes Trace:256  Default Packet Bytes Traced:100
Trace File Record Size:2048  Stop Trace Event: TCP.013
Maximum Hours to HD Shadow:  1
```

Nodisplay

Use the **nodisplay** command to select and turn off messages displaying on the console monitor.

Syntax: <u>nod</u>isplay <u>e</u>vent . . .

                          group . . .

                          <u>s</u>ubsystem . . .

event *subsystem.event#*

    Suppresses the displaying of messages for the specified event.

    Example: `nodisplay event gw.019`

group *group.name*

    Suppresses the displaying of messages that were previously added to the specified group (*group.name*).

    Example: `nodisplay group groupb`

subsystem *subsystem.name*

    Suppresses the displaying of messages associated with the specified subsystem (*logging level*).

    Example: `nodisplay subsystem tkr`

Notrace

Use the **notrace** command to stop display of selected trace events at the console.

Syntax: <u>notrace</u> <u>event</u>. . .

                    <u>group</u> . . .

                    <u>s</u>ubsystem . . .

event *subsystem.event#*

    Suppresses the display of the specified tracing event. (*subsystem.event#*).

    Example: `notrace event gw.019`

group *groupname*

    Suppresses the display of tracing events related to the specified group (*groupname*).

    Example: `notrace group groupb`

subsystem *subsystemname*
> Suppresses the display of tracing events that are associated with the specified subsystem.

> **Example:** `notrace subsystem atm error`
>
> `notrace subsystem atm`

Notrap

See "Notrap" on page 8-13 for an explanation of the **Notrap** command.

Remove

Use the **remove** command to free up memory by erasing stored information. If you have previously saved the current configuration with the **save** command, remove allows you to erase the saved configuration.

Syntax: <u>rem</u>ove

Example: `remove`

Restore

Use the **restore** command to clear all current settings (except counters) and reload the initial ELS configuration. To retain the current settings, use the **save** command before restoring the initial configuration.

Syntax: <u>res</u>tore

Example: `restore`

Retrieve

Use the **retrieve** command to reload the saved ELS configuration. If you have previously saved the current configuration with the **save** command, use **retrieve** to reload it. **Retrieve** does not erase the saved configuration after it executes. To erase the saved configuration, use the **remove** command.

Syntax: <u>ret</u>rieve

Example: `retrieve`

Save

Use the **save** command to store the current configuration (except counters). **Save** does not affect the default configuration (the one you set with the configuration commands). Use **save** after modifying the configuration with the console commands with the intention of saving this configuration over a restart. There can be only one saved configuration at a time. To reload the saved configuration, use the **retrieve** command.

Syntax: <u>s</u>ave

Example: `save`

Set

Use the **set** command to set the maximum number of traps per second, to set the timestamp feature, or to set the tracing options.

pin

    Use the **set pin** command to set the pin parameter to the maximum number of traps that can be sent on a per-second basis. Internally, the pin resets every tenth of a second. (One tenth of the number *max_traps* is sent every tenth of a second.)

    Syntax: <u>s</u>et   *pin max_traps*

    Example: `set pin 100`

timestamp

    Allows you to turn on message timestamping so that either the time of day or uptime (number of hours, minutes, and seconds, but no date, since the router was last initialized) appears next to each message, or to turn off message timestamping.

    Note: If you turn on timestamping, you must remember to go back into the CONFIG process and set the router's date and time using the time command. Otherwise, all messages will come out with 00:00:00, or negative numbers in the hours, minutes, and/or seconds, for example 00:-4:-5.

    Use the **set timestamp** command to enable one of the following timestamp options:

    timeofday   Adds an HH:MM:SS prefix to each ELS message indicating the time of the occurrence during a 24-hour day.

    uptime   Adds an HH:MM:SS prefix to each ELS message indicating the time of the occurrence during a 100-hour cycle of uptime for the router. After 100 hours of uptime, the uptime counter returns to zero to begin another 100-hour cycle.

    off   Turns off the ELS timestamp prefix.

    Syntax: <u>s</u>et timestamp   timeofday OR uptime OR off

    Example: `set timestamp timeofday`

trace

    Use the **set trace** command to configure tracing options. When tracing options are configured from the monitoring console, the changes take effect immediately, and return to their previously configured settings when the device is rebooted.

    Syntax: <u>s</u>et trace   decode . . .
                               default-bytes-per-port . . .
                               disk-shadowing . . .
                               max-bytes-per-port . . .
                               memory-trace-buffer-size . . .
                               off
                               on
                               reset
                               stop-event . . .
                               wrap-mode . . .

decode *off / on*
> Turns packet decoding on or off. Packet decoding is not supported by all components.

default-bytes-per-pkt *bytes*
> Sets the default number of bytes traced. This value is used if a value is not specified by the component doing the tracing.

disk-shadowing *off / on* OR *record-size* OR *time-limit*
> Turns disk shadowing on or off, sets the maximum trace file size, or sets the maximum time for disk-shadowing traces.

> off / on
>> Turns disk shadowing on or off. If disk shadowing is enabled, trace records are copied to the hard disk. Once a traced record is copied to the hard disk, it can no longer be viewed from the console.
>>
>> **Note:** Disk shadowing should be set to OFF whenever the WRITE, TFTP software, RETRIEVE system dump, or COPY software commands are issued.
>>
>> **Example:**   `set trace off`
>>
>> Turns disk shadowing on or off and sets the maximum trace file size. If disk shadowing is enabled, trace records are copied to the hard disk. Once a traced record is copied to the hard disk, it is no longer viewable through the console.

> record-size
>> Sets the record size for trace file records:
>>
>> **Valid Values:** 1024, 2048, or 4096 bytes
>> **Default:**   2048 bytes
>>
>> **Notes:**
>>
>> 1. If a trace file already exists, "Cannot change Record Size without first deleting the existing Trace File" is displayed and record size is not changed.
>>
>> 2. If you configure a record size and a trace file already exists, the trace will use the record size of the existing file.
>>
>> **Example:**   `set trace record-size 4096`

> delete-file
>> Deletes the trace file (in the subdirectory associated with the active bank only).
>>
>> **Note:** If disk shadowing is ON when the command is issued, " Disk-shadowing must be set to OFF before trace file can be deleted" is displayed and the file is not deleted.

> time-limit
>> Sets the maximum time for disk-shadowing of traces:
>>
>> **Valid Values:** 1 - 72 hours:
>> **Default**   24 hours
>>
>> **Note:** Disk shadowing stops (tracing continues) after this time has elapsed. The actual time is reset to 0 when disk shadowing is turned on again.
>>
>> **Example:**   `set trace disk-shadowing time 36`

max-bytes-per-pkt *bytes*
> Sets the maximum number of bytes traced for each packet.

memory-trace-buffer-size *bytes*
> Sets the size, in bytes, of the RAM trace buffer.

off
> Disables packet tracing.

on
> Enables packet tracing.

reset
> Clears the trace buffer and resets all associated counters.

stop-event *event id*
> Stops tracing when an event (event id) occurs. Enter either an ELS event id (for example: TCP.013) or "None." "None" is the default. Tracing stops only if the display of the particular ELS event is enabled.
>
> When a stop-event occurs, an entry is written to the trace buffer. The **view** command for this trace entry will display "Tracing stopped due to ELS Event Id: TCP.013."
>
> After tracing stops due to a stop-event, you must re-enable tracing with the **set trace on** command. (A restart will also re-enable tracing if enabled from the ELS Config> prompt.)
>
> **Example:** `set trace stop-event TCP.013`

wrap-mode *off/on*
> Turns the trace buffer wrap mode on or off. When wrap mode is enabled and the trace buffer is full, previous trace records will be overwritten by new trace records as necessary to continue tracing.

Example $_1$: `set trace decode on`

Example $_2$: `set trace default-bytes-per-packet 64`

Example $_3$: `set trace off`

Statistics

Use the **statistics** command to display a list of all of the available subsystems and their statistics.

Note: The following example may not match your display exactly. The output of the command depends on the version and release of the installed software.

Syntax: <u>st</u>atistics

Example: `statistics`

Subsys	Vector	Exist	String	Active	Heap
GW	80	78	2376	3	36
FLT	20	7	184	0	0
BRS	50	4	161	0	0
ARP	25	19	567	0	0
IP	100	86	2230	0	0
ICMP	30	20	506	0	0
TCP	60	55	2353	0	0
UDP	10	4	109	0	0
BTP	10	5	137	0	0
RIP	30	19	395	0	0
OSPF	80	61	2339	0	0
MSPF	40	15	514	0	0
TFTP	25	24	592	0	0
SNMP	30	20	580	0	0
DVM	30	21	589	0	0
IPX	100	89	3771	0	0
AP2	80	68	1733	0	0
ZIP2	60	51	1859	0	0
R2MP	50	38	1233	0	0
VIN	90	79	3205	0	0
SRT	120	87	4663	0	0
STP	60	32	1590	0	0
BR	50	29	1583	0	0
SRLY	30	28	1448	0	0
ETH	60	47	1098	0	0
TKR	60	45	2031	0	0
X25	70	53	1893	0	0
SDLC	100	92	4105	0	0
FRL	52	52	3127	0	0
PPP	120	93	2939	0	0
X251	50	16	544	0	0
X252	50	34	996	0	0
X253	50	39	1420	0	0
IPPN	20	4	132	0	0
WRS	40	22	1340	0	0
LLC	60	33	1522	0	0
MCF	15	9	239	0	0
DLS	360	337	17148	0	0
V25B	30	28	1058	0	0
BAN	30	29	1223	0	0
Total	2907	2237	83599	3	36

```
Maximum: 3493 vector, 110 subsystem
Memory: 31437/440 vector + 37164/84816 data + 36 heap = 153893
```

Subsys	Name of subsystem
Vector	Maximum size of subsystem
Exist	Number of events defined in this subsystem
String	Number of bytes used for message storage in this subsystem
Active	Number of active (displayed, trapped, or counted) events in the subsystem
Heap	Dynamic memory in use by subsystem

Trace

Use the **trace** command to select the trace events to be displayed on the system console.

Syntax: <u>t</u>race   <u>e</u>vent . . .
                    <u>g</u>roup . . .
                    <u>s</u>ubsystem . . .

event *subsystem.event#*

    Causes the specified trace event (*subsystem.event#*) to be displayed on the system console.

    Example: `trace event gw.019`

group *groupname*

    Allows trace events that were previously added to the specified group to be displayed on the router console.

    Example: `trace group groupb`

subsystem *subsystemname*

    Allows trace events associated with the specified subsystem to be displayed on the router console.

    Example: `trace subsystem gw`

Trap

See "Trap" on page 8-16 for an explanation of the **Trap** command.

View

Use the **view** command to view traced packets.

Syntax: <u>v</u>iew   <u>c</u>urrent
                    <u>f</u>irst
                    <u>j</u>ump
                    <u>l</u>ast
                    <u>n</u>ext
                    <u>p</u>rev
                    <u>s</u>earch ...

current

    Displays the current trace packet. If the current packet is not valid, the first packet in the trace buffer is displayed.

first

    Displays the first traced packet in the trace buffer.

jump *n*

    Displays the traced packet *n* packets ahead of or behind the current packet.

last

    Displays the last traced packet in the trace buffer.

next

    Displays the next traced packet.

prev

    Displays the previous traced packet.

search *hexstring*
Displays the next traced packet that contains the specified hex string.

Example: view current

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: <u>ex</u>it

Example: exit

Chapter 10. Using and Configuring Bandwidth Reservation (BRS) and Priority Queuing

This chapter describes the bandwidth reservation and priority queuing features currently available for Frame Relay and PPP interfaces, and the commands for configuring them. It includes the following sections:

- "Bandwidth Reservation System (BRS)"
- "Priority Queuing" on page 10-2
- "Bandwidth Reservation with Priority Queuing" on page 10-3
- "Bandwidth Reservation for Traffic Classes" on page 10-3
- "Bandwidth Reservation over Frame Relay" on page 10-5
- "Accessing the Bandwidth Reservation Configuration Prompt" on page 10-6
- "Bandwidth Reservation Configuration Commands" on page 10-8

Bandwidth Reservation System (BRS)

The Bandwidth Reservation System (BRS) allows you to decide which packets to drop when demand (traffic) exceeds supply (throughput) on a network connection. Bandwidth reservation is not used until more than 100% of the available line bandwidth is requested.

Bandwidth reservation "reserves" transmission bandwidth for a network connection. This reservation feature allocates minimum percentages of total connection bandwidth for specified *classes* of traffic. See Figure 10-1.
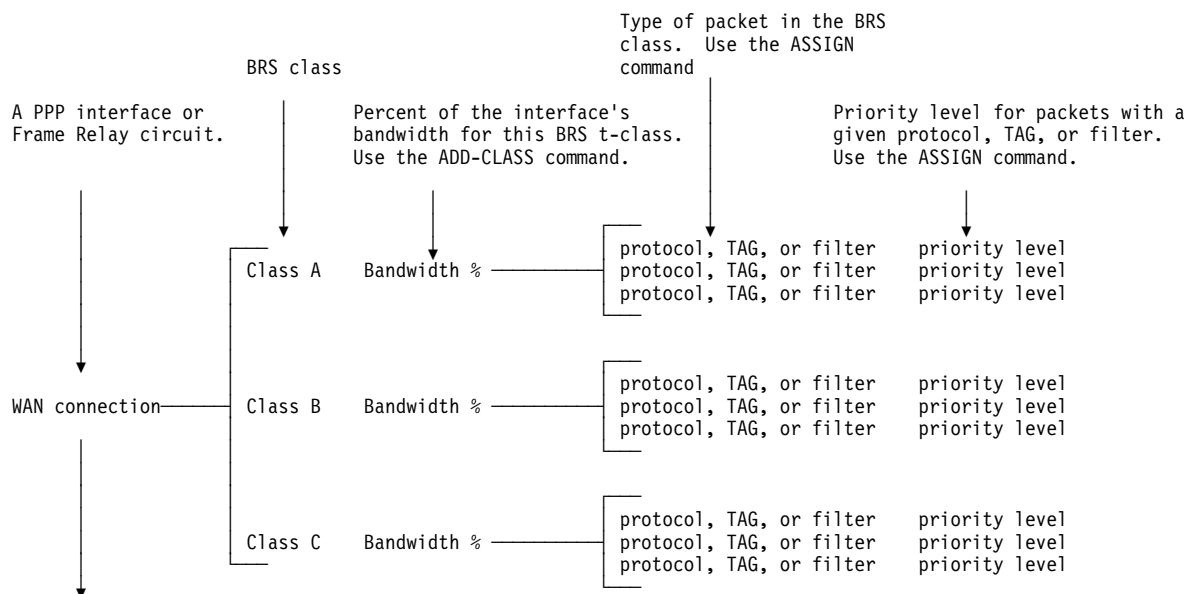
Figure 10-1. Components of Bandwidth Reservation System (BRS) Classes

These reserved percentages are a minimum *slice* of bandwidth for the network connection. If a network is operating to capacity, messages in any one class can

10-1

be transmitted only until they use the configured bandwidth allocated for the class. In this case, additional transmissions are held until other bandwidth transmissions have been satisfied. In the case of a light traffic path, a packet stream can use bandwidth exceeding its allowed minimum up to 100% if there is no other traffic.

Bandwidth reservation is really a *safeguard*. In general, a network should not attempt to use greater than 100% of its line speed.   If it does, a faster line is probably needed. The "bursty" nature of traffic, however, can drive the requested transmission rate to exceed 100% for a short time. In these cases, bandwidth reservation is enabled and the higher priority traffic is ensured delivery (that is, is not discarded).

Bandwidth reservation runs over the following datalink types:

- Frame Relay (serial line or dial circuit interface)
- PPP (serial line or dial circuit interface)

Priority Queuing

Bandwidth reservation allocates percentages of total connection bandwidth for specified traffic *classes*, or *t-classes*, defined by the user. A BRS t-class is a group of packets identified by the same name; for example, a class called "ipx" to designate all IPX packets.

With priority queuing, each bandwidth t-class can be assigned one of the following priority level settings:

- Urgent
- High
- Normal (the default setting)
- Low

All packets assigned the Urgent priority are sent first within their class. These packets are followed by High, Normal, and then Low messages respectively. When all Urgent packets have been transmitted, High packets are transmitted until all are sent (or until new Urgent messages are queued). Only when there are no Urgent, High, or Normal packets remaining are the Low priority packets transmitted. If no priority setting is assigned, the setting defaults to Normal.

Also, you can set the number of packets that are waiting in the queue for each priority level in each bandwidth t-class. The BRS **queue-length** command sets the maximum number of output buffers that can be queued in each BRS priority queue, and the maximum number of output buffers that can be queued in each BRS priority queue for when router input buffers are scarce. You can set up priority queue lengths for both PPP and Frame Relay.

Attention: If you set the values for queue length too high, you may seriously degrade the performance of your router.

For BRS, you can set priority queue lengths for PPP and Frame Relay WAN interfaces. See "Queue-length" on page 10-16 for a description of the **queue-length** command.

The priority settings in one bandwidth t-class have no effect on other bandwidth classes. No one bandwidth class has priority over the others.

Bandwidth Reservation with Priority Queuing

When priority queuing is configured without bandwidth reservation, the highest priority traffic is delivered first. In instances of heavy high-priority traffic, lower priority levels can be overlooked. By combining priority queuing with bandwidth reservation, however, packet transmission can be allocated to all types of traffic.

Bandwidth Reservation for Traffic Classes

You create a traffic class using the **add-class** command and then assign types of traffic to the class using the **assign** command. Traffic is assigned to a traffic class based on its *protocol type* or based on a filter that further identifies a specific type of *protocol traffic* (for example, SNMP IP packets).

Supported protocol types are:

- IP
- ARP
- DNA
- VINES
- IPX
- OSI
- AP2
- ASRT
- SNA/APPN-ISR
- APPN-HPR

Filtering and Bandwidth Reservation

Using bandwidth reservation, you can assign filters (using the **assign** command) to specific types of traffic. The following filters can be assigned:

- IP tunneling
- SDLC tunneling over IP (SDLC Relay)
- Rlogin
- Telnet
- SNA/APPN-ISR
- APPN-HPR
- SNMP
- IP Multicast
- DLSw
- MAC Address (through MAC filtering tags)
- MAC Filters
- NetBIOS
- Network-HPR
- High-HPR
- Medium-HPR
- Low-HPR
- XTP

MAC Address Filtering and Tags

MAC Address filtering is handled by a joint effort between bandwidth reservation and MAC filtering (MCF) using *tags*. For example, a user with bandwidth reservation is able to categorize bridge traffic by assigning a tag to it.

The tagging process is done by creating a filter item in the MAC filtering configuration console and then assigning a tag number to it. This tag number is used to set up a traffic class for all packets associated with this tag. Tag values must currently be in the range 1 through 64.

Note: Tags can be applied *only* to bridged packets. On a PPP or Frame Relay interface, up to five tagged MAC filters can be assigned as bandwidth reservation filters and are designated as TAG1 through TAG5. TAG1 is searched for first, then TAG2, and so on up to TAG5. A single MAC filter tag can consist of any number of MAC Addresses set in MCF.

Once you have created a tagged filter in the MAC filtering configuration process, you can use the BRS tag configuration command to assign a BRS tag name (TAG1, TAG2, TAG3, TAG4, or TAG5) to the MAC filter tag number. Then use the BRS tag name on the BRS assign command to assign the corresponding MAC filter to a bandwidth traffic class and priority.

Tags also can refer to "groups," as in the example of IP Tunnel. IP Tunnel endpoints can belong to any number of groups. Packets are assigned to a particular group through the tagging feature of MAC Address filtering. For additional information on MAC filtering, refer to Chapter 12, "Using and Configuring MAC Filtering" and Chapter 13, "Monitoring MAC Filtering."

To apply bandwidth reservation and queuing priority to tagged packets:

1. Use the MAC filtering configuration commands at the `filter config>` prompt to set up tags for packets passing through the bridge. See Chapter 12, "Using and Configuring MAC Filtering" for more information.

2. Use the bandwidth reservation **tag** command to reference a tag for bandwidth reservation.

3. With the bandwidth reservation **assign** command, assign the BRS tag to a t-class. The **assign** command also prompts you for a queuing priority within that BRS t-class.

SNA and APPN Filtering

The SNA/APPN-ISR filter allows you to assign SNA and APPN-ISR traffic that is being bridged to a BRS traffic class. SNA and APPN-ISR traffic is identified as any bridge packets with a destination or source SAP of 0x04, 0x08, or 0x0C and whose LLC (802.2) control field indicates that it is not an unnumbered information (UI) frame.

Note: Frame Relay BAN packets are this category.

The APPN-HPR filters allow you to assign HPR traffic that is being bridged to a BRS t-class. HPR traffic is identified as any bridge packet with a destination or source SAP of 0x04, 0x08, 0x0C, or 0xC8 and whose LLC (802.2) control field indicates it is an unnumbered information (UI) frame.

The Network-HPR, High-HPR, Medium-HPR, and Low-HPR filters allow HPR bridge traffic to further be filtered according to the HPR transmission priority. For example, if you want to assign HPR traffic that uses the network transmission priority to one t-class and priority and all other HPR bridged traffic to a different t-class or priority, you would assign the Network-HPR filter to the appropriate t-class and priority and use the APPN-HPR filter to assign the rest of the HPR traffic to a different t-class or priority.

Other filters may help you to assign traffic. For example, the DLSw filter allows you to assign SNA-DLSw traffic that is being sent over a TCP connection to a BRS t-class.

For SNA/APPN-ISR and APPN-HPR filters, if you want to check for SAPs other than the ones above, create a sliding window filter using MAC filtering and tag that filter. Then assign the tagged MAC filter to a BRS t-class.

Order of Precedence

It is possible for a packet to fall under several filterable classes. For example, an IP Tunneled bridged packet for SNA with a filter for a MAC Address. The order for resolving the filtering priority for this packet would be as follows:

1. MAC Address match for bridging (IP/ASRT) tag 1 to tag 5
2. NetBIOS for bridging (IP/ASRT)
3. SNA/APPN-ISR for bridging (IP/ASRT)
4. HPR-Network (IP/ASRT)
5. HPR-High (IP/ASRT)
6. HPR-Medium (IP/ASRT)
7. HPR-Low (IP/ASRT)
8. APPN-HPR (IP/ASRT)
9. IP tunneling (IP)
10. SDLC relay (IP)
11. Multicast (IP)
12. SNMP (IP)
13. Rlogin (IP)
14. Telnet (IP)
15. DLSw (IP)
16. XTP (IP)

Bandwidth Reservation over Frame Relay

Bandwidth reservation allows you to reserve bandwidth at two levels:

- At the interface level, you can assign a percentage of the interface's bandwidth to circuit classes (*c-classes*). Each circuit class contains one or more circuits.

- At the circuit level, you can define traffic classes and allocate a percentage of the circuit's bandwidth.

Packets are filtered and queued into BRS t-classes based on the packet's protocol type and any configured BRS filters. The packets are then queued into a BRS c-class based on the DLCI number.

The actual amount of bandwidth available for bandwidth reservation depends upon how you configure the interface and circuit:

- If you enable Frame Relay CIR monitoring, the bandwidth available to the circuit is allocated strictly according to its committed information rate (CIR), its committed burst size, and its excess burst size.

- If you disable CIR monitoring, up to 100 % of the bandwidth of the interface may be available to a circuit.

Orphaned circuits and circuits without BRS explicitly enabled use a default BRS queuing environment where the packets are queued on the default t-class and priority and the default c-class.

You can use several bandwidth reservation monitoring commands to display reservation counters for the circuit classes for a given interface:

- clear-circuit-class
- counters-circuit-class
- last-circuit-class

See Chapter 11, "Monitoring Bandwidth Reservation" for more information on monitoring BRS.

The interface is the one shown at your prompt for the bandwidth monitoring commands. For example, `BRS [i 5]` is the prompt for interface 5.

If you want to use BRS circuit classes, leave all circuits in the default c-class and do not create any other circuit classes.

Queuing Support

With bandwidth reservation over Frame Relay, each circuit can queue frames while in the congested state, even for interfaces and circuits that are not enabled for bandwidth reservation.

Accessing the Bandwidth Reservation Configuration Prompt

To access bandwidth reservation configuration commands and configure bandwidth reservation on your router:

1. At the `OPCON` (*) prompt, enter **talk 6**.

2. At the `Config>` prompt, enter **feature brs**.

3. At the `BRS Config>` prompt, enter **interface #**.

4. At the `BRS [i 0] Config>` prompt, enter **enable**.

   This is the interface prompt level, and the interface number is zero in this instance. You need to repeat step 3 and step 4 for each interface you are configuring.

   If you are configuring BRS on a Frame Relay interface, continue with step 4a:

   If you are configuring BRS on any other interface, go directly to step 5.

   a. At the `BRS [i 0] Config>` prompt, enter **circuit #**, where # is the number of the circuit you want to configure.

   b. At the `BRS [i 0] [dlci 16] Config>` prompt, enter **enable**. This is the circuit prompt level and the circuit (DLCI) number is 16 in this instance.

       c. At the `BRS [i 0] [dlci 16] Config>` prompt, enter **exit** to return to the interface level prompt.

       d. Repeat steps 4a through 4c for each circuit for which you want to define BRS t-classes.

5. Restart your router.

6. Repeat steps 1 through 3 to configure bandwidth reservation for the particular interface that you have enabled.

7. At the `BRS [i 0] Config>` prompt, configure the bandwidth reservation parameters for the selected interface by using the appropriate configuration commands discussed in this chapter. For Frame Relay interfaces, at the `BRS [i 0] Config>` prompt, type **circuit** *permanent-virtual circuit #*. This will access the DLCI prompt level for configuring the BRS t-class parameters.

Note: You do not need to restart the router for t-class and c-class configuration changes to take effect.

The **talk 6 (t 6)** command lets you access the configuration process.

The **feature brs** command lets you access the BRS configuration process. You can enter this command by using either the feature name (brs) or number (1).

The **interface #** command selects the particular interface that you want to configure for bandwidth reservation. Before configuring any BRS classes, you must use the **enable** command to enable BRS on the interface. In Step 4 on page 10-6, the prompt indicates that the selected interface's number is zero.

The **circuit #** command selects the circuit on the FR interface on which you want to configure BRS traffic classes. Before configuring any BRS t-classes for the circuit, you must use the **enable** command to enable BRS on the circuit. In step 4b on page 10-6, the prompt indicates that circuit 16 on interface 0 has been selected.

You must enable bandwidth reservation for the selected interface and circuit and then restart your router before configuring circuit classes (Frame Relay only), and traffic classes.

To return to the `Config>` prompt at any time, enter the **exit** command at the different levels of BRS prompts until you are at the `Config>` prompt.

Bandwidth Reservation Configuration Commands

This section describes the Bandwidth Reservation configuration commands. The commands that can be used differ depending on the BRS configuration prompt that is displayed (`BRS Config>`, `BRS [i x] Config>`, or `BRS [i x] [dlci y] Config>`). See "? (Help)" on page 10-10 for a list of the commands supported at each BRS configuration prompt.

The commands marked with a check (yes) are used only with Frame Relay.

Command	Used Only With FR	Function
Table 10-1 (Page 1 of 2). Bandwidth Reservation Configuration Command Summary		
? (Help)		Displays the Bandwidth Reservation configuration commands or lists the subcommand options for specific commands (if available).
Add-circuit-class	yes	Sets the name of a bandwidth c-class and its percentage of bandwidth.
Add-class		Allocates a designated amount of bandwidth to a user-defined traffic class.
Assign		Assigns a protocol or filter to a configured traffic class.
Assign-circuit	yes	Assigns a specified circuit to the specified bandwidth c-class.
Change-circuit-class	yes	Changes the percentage of the bandwidth to be used by the group of circuits assigned to the designated c-class.
Change-class		Changes the amount of bandwidth configured for a bandwidth t-class.
Circuit	yes	Selects the DLCI of a Frame Relay permanent virtual circuit.
Clear-block		When issued at the circuit level, clears the configuration data associated with the current circuit from SRAM. When issued at the interface level, clears the configuration data associated with the current interface from SRAM.
Deassign		Restores the queuing of the specified packet or filter to the default t-class and priority.
Deassign-circuit	yes	Restores the specified circuit to the default c-class
Default-circuit-class	yes	Assigns the name of a default bandwidth c-class and its percentage of the interface's bandwidth.
Default-class		Sets the default t-class and priority to a desired value.
Del-circuit-class	yes	Deletes the specified bandwidth c-class.
Del-class		Deletes a previously configured bandwidth t-class.
Disable		Disables bandwidth reservation on the interface if entered from the interface prompt, or on the circuit if entered from the circuit prompt.

Table 10-1 (Page 2 of 2). Bandwidth Reservation Configuration Command Summary

Command	Used Only With FR	Function
Enable		Enables bandwidth reservation on the interface if entered from the interface prompt or on the circuit if entered from the circuit prompt.
Interface		Selects an interface on which to run bandwidth reservation. **Note:** This command must be entered before using any other configuration commands.
List		At the interface prompt for PPP and the circuit prompt for Frame Relay, displays the configured t-classes and assigned protocols and filters. At the interface prompt for Frame Relay, displays the c-classes and assigned circuits.
Queue-length		Sets the maximum and minimum values for the number of packets in a priority queue.
Show		Displays the currently defined bandwidth classes stored in RAM. At the interface prompt for PPP and the circuit prompt for Frame Relay, it displays the t-classes. At the interface prompt for Frame Relay, it displays the c-classes.
Tag		Assigns a BRS tag name (TAG1-TAG5) to a filter that has been tagged during the configuration of the MAC Filtering feature.
Untag		Removes the relationship between a BRS tag name (TAG1-TAG5) and a filter that has been tagged during configuration of the MAC filtering feature.
Exit		Exits the current bandwidth reservation prompt.

Use the appropriate commands to configure bandwidth reservation for the Point-to-Point protocol (PPP) and Frame Relay. For Frame Relay, you need to configure the circuit and the network interface. For PPP, you only need to configure the network interface.

Notes:

1. When the **clear-block, disable**, **enable**, **list**, and **show** commands are issued from within the BRS interface menu, they affect or list the bandwidth reservation information configured for the selected interface. When these commands are issued from within the BRS circuit menu, only the Frame Relay bandwidth reservation information configured for the permanent virtual circuit (PVC) is affected or listed.

2. Before using the bandwidth reservation commands, keep the following in mind:

   - You must use the **interface** command to select an interface before you use any other configuration commands. (BRS configuration enforces this.)

   - The *Class-name* parameter is case-sensitive.

   - To view the current *class-names*, use the **list** or **show** command.

- After you enable bandwidth reservation on an interface or circuit, you can add/delete/change circuit and traffic classes and assign circuits or protocols dynamically. The only commands that require a router restart before taking effect are the enable, disable, and clear-block commands.

Note: You do not need to restart the router for t-class and c-class configuration changes to take effect.

? (Help)

Use the **?** (**help**) command to list the available commands from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

The following example shows the output that is displayed at the BRS [i #] Config> prompt when you are configuring for Frame Relay.

```
?(HELP)
ENABLE
DISABLE
CIRCUIT
ADD-CIRCUIT-CLASS
DEL-CIRCUIT-CLASS
CHANGE-CIRCUIT-CLASS
DEFAULT-CIRCUIT-CLASS
ASSIGN-CIRCUIT
DEASSIGN-CIRCUIT
LIST
QUEUE-LENGTH
SHOW
CLEAR-BLOCK
EXIT
```

The following example shows the output that is displayed at the BRS [i #] Config> prompt for non-Frame-Relay interfaces, and at the BRS [i #] [dlci #] Config> prompt for Frame-Relay interfaces.

```
?(HELP)
ENABLE
DISABLE
ADD-CLASS
DEL-CLASS
CHANGE-CLASS
DEFAULT-CLASS
TAG
UNTAG
ASSIGN
DEASSIGN
LIST
QUEUE-LENGTH
SHOW
CLEAR-BLOCK
EXIT
```

At the BRS Config> prompt, the following commands are listed:

```
?(HELP)
INTERFACE
LIST
EXIT
```

Add-circuit-class

Used only when configuring Frame Relay.

Use the **add-circuit-class** command at the interface level to allocate a designated amount of bandwidth to be used by the group of circuits assigned to the user-defined bandwidth c-class.

Syntax: <u>ad</u>d-circuit-class *class-name %*

Example: `add-circuit-class alpha 10`

Add-class

Use the **add-class** command to allocate a designated amount of bandwidth to a user-defined bandwidth t-class.

Syntax: <u>ad</u>d-class *class-name* or *class#*

Example: `add test 20`

Assign

Use the **assign** command to assign specified tags, protocol packets, or filters to a given t-class and priority within that class. The four priority types include:

- Urgent
- High
- Normal (the default priority)
- Low.

Syntax: <u>as</u>sign   *protocol-class or TAG or filter-class   class-name or class#*

Example: **`assign IPX test`**
    `priority <URGENT/HIGH/NORMAL/LOW>: [NORMAL]? low`
    `protocol IPX maps to class test with priority LOW`

Assign-circuit

Used only when configuring Frame Relay.

Use the **assign-circuit** command at the interface level to assign the specified circuit (DLCI) to the specified bandwidth c-class.

Note: You must use the **circuit** command to enable BRS on the DLCI and restart the router before you can use this command to assign the circuit to a circuit class.

Syntax: <u>as</u>sign-circuit *# class name*

Example: `assign-circuit 16 pubs`

Change-circuit-class

Used only when configuring Frame Relay.

Use the **change-circuit-class** command at the interface level to change the percentage of the bandwidth to be used by the group of circuits assigned to the specified c-class.

Syntax: <u>ch</u>ange-circuit-class *class-name %*

Example: `change-circuit-class alpha 20`

Change-class

Use the **change-class** command to change the amount of bandwidth configured for a bandwidth t-class.

Syntax: <u>ch</u>ange-class *class-name or class# %*

Example: `change test 10`

Circuit

Used only when configuring Frame Relay.

Use the **circuit** command to configure the DLCI of a Frame Relay permanent virtual circuit (PVC). This command can only be issued from the BRS interface configuration prompt (`BRS [i #] Config>`).

Syntax: circuit *permanent-virtual-circuit #*

Example: `circuit 16`

Before you can use the **add-class**, **assign**, **default-class**, **del-class**, **deassign**, or **changet-class** commands, you must enable BRS on the circuit and restart the router. For example.

```
BRS [i 1] Config> circuit
Circuit to reserve bandwidth:  [16]

BRS [i 1 ] [dlci 16] Config> enable
```

After the **enable** command is issued for the Frame-Relay circuit and the router is restarted, the following configuration commands are available for the circuit:

add-class	deassign	enable	tag
assign	default-class	exit	untag
change-class	del-class	list	
clear-block	disable	show	

Clear-block

Use the **clear-block** command to clear the current bandwidth reservation configuration data from SRAM.

- If you enter this command from the interface prompt for PPP, all BRS configuration data is cleared for the interface.

- If you enter this command from the interface prompt for Frame Relay, BRS is no longer enabled on the interface or on any circuits of the interface, and all circuit-class configuration data is cleared. However, the traffic-class configuration data for each individual circuit is not cleared and is available if you re-enable BRS on the interface.

- To clear a circuit's traffic-class configuration data, you first enter the **circuit** command from the interface-level prompt and then the **clear-block** command from the circuit-level prompt. After you have cleared the traffic-class configuration data for each circuit, enter the **clear-block** command from the interface-level prompt to clear the circuit-class configuration data. The changes do not take effect until the router is restarted.

Syntax: clear-block

Example: `clear-block`

             `You are about to clear BRS configuration information for this interface`
             `Are you sure you want to do this (Yes or No):` **`y`**
             `BRS [i 1] Config>`

Default-circuit-class

Used only when configuring Frame Relay.

Use the **default-circuit-class** command at the interface level to set the user-defined name of the default bandwidth c-class and the percentage of the bandwidth allocated to that class of circuits, including orphans, that are not assigned to a bandwidth c-class.

Syntax: <u>def</u>ault-circuit-class *class-name %*

Example: `default-circuit-class group 10`

Del-circuit-class

Used only when configuring Frame Relay.

Use the **del-circuit-class** command at the interface level to delete the specified bandwidth c-class.

Syntax: <u>del</u>-circuit-class *class-name*

Example: `del-circuit-class group`

Default-class

Use the **default-class** command to set the default t-class and priority to a desired value. If no value has been previously assigned, system default values are used. Otherwise, the last previously assigned value is used.

Syntax: <u>def</u>ault-class *class-name* or *class# priority*

Example: `default-class test normal`

Del-class

Use the **del-class** command to delete a previously configured bandwidth t-class from the specified interface or circuit.

Syntax: del-class *class-name* or *class#*

Example: `del-class ip`

Deassign

Use the **deassign** command to restore the queuing of the specified protocol packet or filter to the default t-class and priority.

Syntax: deassign *prot-class* or *filter-class*

Example: `deassign IP`

Deassign-circuit

Used only when configuring Frame Relay.

Use the **deassign-circuit** command at the interface level to restore the queuing of the specified circuit to the default c-class.

Syntax: deassign-circuit *#*

Example: `deassign 16`

Disable

Use the **disable** command to disable bandwidth reservation on the interface (if entered from the interface prompt) or on the circuit (if entered from the circuit prompt). The changes do not take effect until the router is restarted.

To verify that bandwidth reservation is disabled, enter the **list** command.

Syntax: disable

Example: `disable`

Enable

Use the **enable** command to enable bandwidth reservation on the interface (if entered from the interface prompt) or the circuit (if entered from the circuit prompt). The changes do not take effect until the router is restarted.

Syntax: enable

Example: `enable`

Note:

- When configuring BRS on a PPP interface, issue the **enable** command at the interface prompt, and then restart the router before configuring any traffic classes and assigning protocols and filters to traffic classes.

- When configuring BRS on a Frame-Relay interface, issue the **enable** command at the interface prompt and at the circuit prompt of each circuit for which you

want to define traffic classes. Then restart the router before configuring circuit classes for the interface and traffic classes for each circuit.

Interface

Use the **interface** command to select the serial interface to which bandwidth reservation configuration commands will be applied. *Bandwidth reservation is supported on routers running PPP (Point-to-Point Protocol) and Frame Relay interfaces.*

Notes:

1. To enter bandwidth reservation commands for a new interface, this command must be entered **before** using any other bandwidth reservation configuration commands. If you have exited the bandwidth reservation prompt and wish to return to make bandwidth reservation changes to a previously configured interface, this command must again be entered first.

2. If WAN Restoral is used and BRS is configured on a primary interface, BRS should also be configured on the secondary interface. Typically when WAN Restoral is used, the secondary interface takes on the identity of the primary interface. This is not true for BRS; therefore, BRS needs to be configured on both the primary and secondary interfaces.

To enable Bandwidth Reservation on a particular interface, at the `BRS Config>` prompt, enter the number of the interface that supports the particular protocol or feature. You can then use the BRS **enable** configuration command as described in this chapter. After enabling the interface number, you must restart the 2216 for the command to take effect before you can make any other configuration changes to the interface.

Notes:

1. If you are configuring BRS on a Frame Relay interface, you can use the **circuit** command to select circuits and enable bandwidth reservation on those circuits before you restart the router.

Syntax: interface *interface#*

Example: `interface 2`

List

Use the **list** command to display currently defined bandwidth classes and their guaranteed percentage rates.

The **list** command is very similar to the **show** command. However, the **show** command displays the current settings from the active RAM display.

Syntax: list *interface#*

Example: `list`

Depending on the prompt at which you issue the **list** command, various outputs are displayed. You can issue the **list** command from the following prompts:

```
                          BRS [i 2] Config>
                          BRS [i 1] [dlci 16] Config>
                          BRS [i 1] Config>
                          BRS Config>
```

The following output is displayed when you issue the **list** command at the BRS
Config> prompt:

```
          Bandwidth Reservation is available for 2 interfaces.

          Interface   Type    State
          ---------   ----    -----
                  1    FR      Enabled
                  2    PPP     Enabled
```

At the BRS interface level prompt (BRS [i 0]) for PPP interfaces and at the BRS
circuit level prompt (BRS [i 0] [dlci 16] Config>) for Frame Relay interfaces, the
list command lists the traffic classes, their configured bandwidth percentages, and
the assigned protocols and filters.

At the BRS interface level prompt for Frame Relay, the **list** command lists the
circuit classes, their configured bandwidth percentages, and the assigned circuits.

Queue-length

Use the **queue-length** command to set the number of packets that can be queued
in each BRS priority queue. Each BRS class has a priority value assigned to its
protocols, filters, and tags, and each priority queue can store the number of packets
that you specify with this command.

This command sets the maximum number of buffers that can be queued in each
BRS priority queue as well as the maximum number that can be queued in each
BRS priority queue when there is a shortage of router input buffers.

If you issue **queue-length** for a PPP interface or a dial circuit, the command sets
the queue-length values for each priority queue of each BRS t-class that is defined
for the interface.

If you issue **queue-length** for a Frame Relay interface (at the prompt: BRS [i
0] Config>), the command sets the default queue-length values for each priority
queue of each BRS t-class that is defined for each permanent virtual circuit of the
interface.

If you issue **queue-length** for a Frame-Relay PVC (at a prompt like this: BRS [i
0] [dlci 16] Config>) the command sets the queue length values for each priority
queue of each BRS t-class that is defined for the PVC. These values override the
default queue length values set for the Frame Relay interface.

Attention: Do not use this command unless it is essential to do so. The default
values for queue length are the recommended values for most users. If you set the
values for queue length too high, you may seriously degrade the performance of
your router.

Syntax: queue-length *maximum-length*　*minimum-length*

Example: `queue-length`

```
BRS priority queue maximum length [10]?
BRS priority queue minimum length [3]
```

Show

Use the **show** command to display currently defined bandwidth classes stored in RAM.

Syntax: show *interface#*

Example: `show`

Depending on the prompt at which you issue the **show** command, various outputs are displayed. You can issue the **show** command from the following prompts:

> BRS [i *x*] Config> - interface level prompt for interface number *x*.
> BRS [i *x*] [dlci *y*] Config> - circuit level prompt for circuit *y* on Frame Relay interface number *x*.

At the interface prompt for PPP and the circuit prompt for Frame Relay, traffic class information is displayed. At the interface prompt for Frame Relay, circuit class information is displayed.

Tag

Use the **tag** command to assign a filter that has been tagged during the configuration of the MAC filtering feature to the next available BRS tag name. The BRS tag names are TAG1, TAG2, TAG3, TAG4, and TAG5. You use the BRS tag name on the assign command to assign the tag to a BRS traffic class.

Use the **list** command to list which MAC filter tags have been assigned to a BRS tag name and which BRS tag names have been assigned to a bandwidth traffic class.

Syntax: tag *mac_filter_tag#*

Example: `tag 3`

Untag

Use the **untag** command to remove the MAC filter tag number and BRS tag name relationship. A tag can be removed only if its corresponding BRS tag name is not assigned to a bandwidth traffic class.

Use the **list** command to show which MAC filter tags are assigned to a BRS tag name and which BRS tag names are assigned to a traffic class.

Syntax: untag *mac_filter_ag#*

Example: `untag 3`

Exit

Use the **exit** command to return to the previous level prompt. For example, entering the **exit** command at the circuit level prompt (for example, BRS [i 0] [dlci 16] Config>) takes you back to the interface level prompt (BRS [i 0] Config>). Entering the **exit** command at the interface level prompt returns you to the general BRS prompt (BRS Config>), and entering the **exit** command at the general BRS prompt returns you to the Config> prompt.

Syntax: exit

Example: exit

Chapter 11. Monitoring Bandwidth Reservation

This chapter describes how to access the Bandwidth Reservation System (BRS) console prompt and the available console commands.

This chapter includes the following sections:

- "Accessing the Bandwidth Reservation Console Prompt"
- "Bandwidth Reservation Console Commands" on page 11-2

Accessing the Bandwidth Reservation Console Prompt

To access bandwidth reservation console commands and to monitor bandwidth reservation on your router, do the following:

1. At the OPCON prompt (*), type **talk 5**.

2. At the GWCON prompt (+), type **feature brs.**

3. At the BRS> prompt, type **interface #**, where # is the number of the interface that you want to monitor. This takes you to the BRS interface-level prompt, BRS [i x]>, where x is the number of the interface number.

4. For Frame Relay only, type **circuit #** at the interface prompt to specify the circuit on this interface that you want to monitor.

   This takes you to the circuit-level prompt BRS [i x] [dlci y]>, where x is the interface number and y is the circuit number.

5. At the prompt, type the appropriate monitoring command. (Refer to "Bandwidth Reservation Console Commands" on page 11-2.)

   The **talk 5 (t 5)** command lets you access the monitoring process.

   The **feature brs** command lets you access the BRS monitoring process. You can enter this command by using either the feature name (brs) or number (1).

   The **interface #** command selects the particular interface that you want to monitor for bandwidth reservation.

   The **circuit #** command selects the DLCI of a Frame Relay permanent virtual circuit (PVC).

   To return to the GWCON prompt at any time, type the **exit** command at the BRS> prompt.

   Once you access the bandwidth reservation console prompt (BRS>), you can enter any of the specific console commands described in Table 11-1 on page 11-2.

Bandwidth Reservation Console Commands

This section summarizes and explains the Bandwidth Reservation console commands. Table 11-1 shows the Bandwidth Reservation console commands. The commands that can be used differ depending on the BRS console prompt (BRS>, BRS [i x]>, or BRS [i x] [dlci y]>). See "? (Help)" for a list of the console commands supported at each BRS console prompt.

Command	Used Only With FR	Function
Table 11-1. Bandwidth Reservation Console Command Summary		
? (Help)		Displays the Bandwidth Reservation configuration commands or lists the subcommand options for specific commands (if available).
Circuit	yes	Selects the DLCI of a Frame Relay permanent virtual circuit (PVC). To monitor Frame Relay bandwidth reservation traffic, you must be at the circuit prompt level.
Clear		Clears the current t-class counters and stores them as **last** t-class counters. Counters are listed by class.
Clear-circuit-class	yes	Clears the current c-class counters and stores them as **last** c-class counters. Counters are listed by class.
Counters		Displays the current t-class counters.
Counters-circuit-class	yes	Displays the current c-class counters.
Interface		Selects the interface to monitor.
		Note: This command must be entered before using any other bandwidth reservation console commands.
Last		Displays the last saved t-class counters.
Last-circuit-class	yes	Displays the last saved c-class counters.
Exit		Exits the bandwidth reservation console process.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

At the BRS> prompt:

```
INTERFACE
EXIT
```

For Frame Relay, at the BRS [i #] [dlci #]> prompt or for PPP, at the BRS [i #] prompt, enter:

```
CLEAR
COUNTERS
LAST
EXIT
```

For Frame Relay, at the BRS [i #]> prompt, enter: "Bandwidth Reservation Console Commands" on page 11-2.

```
CIRCUIT
CLEAR-CIRCUIT-CLASS
COUNTERS-CIRCUIT-CLASS
LAST-CIRCUIT-CLASS
EXIT
```

Circuit

Used only when monitoring Frame Relay.

Use the **circuit** command to select the DLCI of a Frame Relay PVC for monitoring. This command can be issued only from the BRS interface monitoring prompt (BRS [i #]>).

Syntax: c̲ircuit *permanent virtual circuit #*

Example: `circuit 16`

After the Frame Relay circuit has been selected, the following commands can be used at the circuit prompt:

```
CLEAR
COUNTERS
LAST
EXIT
```

Clear

Use the **clear** command to save the current bandwidth reservation t-class counters so that they can be retrieved using the **last** command and clear the values. The counters are kept on a bandwidth traffic class basis.

Syntax: c̲lear

Example: `clear`

Clear-Circuit-Class

Used only when monitoring Frame Relay.

Use the **clear-circuit-class** command to save the current bandwidth reservation c-class counters so that they can be retrieved using the **last-circuit-class** command and clear the values. The counters are kept on a circuit class basis.

Syntax: c̲lear

Example: `clear-circuit-class`

Counters

Use the **counters** command to display statistics describing bandwidth reservation traffic for the traffic classes configured for a PPP interface or Frame Relay circuit.

Syntax: <u>co</u>unters

Example: `counters`

```
Bandwidth Reservation Counters
Interface 1

Class        Pkt Xmit    Bytes Xmit    Bytes Ovfl
LOCAL               0             0             0
DEFAULT             1            30             0
CLASS 1             1            56             0
CLASS 2             0             0             0

TOTAL               2            86             0
```

Counters-Circuit-Class

Used only when monitoring Frame Relay.

Use the **counters-circuit-class** command to display statistics for the traffic classes configured for a Frame Relay circuit.

Syntax: <u>co</u>unters-circuit-class

Example: `counters-circuit-class`

```
Bandwidth Reservation Circuit Class Counters
Interface 1


Class        Pkt Xmit    Bytes Xmit    Bytes Ovfl
DEFAULT            25          3402            26
CIRCLASS1           1            56             0
CIRCLASS2           0             0             0

TOTAL              26          3458            26
```

Interface

Use the **interface** command to select the serial interface to which bandwidth reservation console commands will be applied. *Bandwidth reservation is supported on routers running the PPP (Point-to-Point Protocol) and Frame Relay interfaces.*

Note: To enter bandwidth reservation commands for a new interface, this command must be entered before using any other bandwidth reservation console commands. If you have exited the bandwidth reservation console prompt (BRS>) and want to return to monitor bandwidth reservation, you must again enter this command first.

To monitor Bandwidth Reservation on a particular interface, at the BRS> console prompt, type the number of the interface. You can then use bandwidth reservation console commands as described in this chapter.

Syntax: interface *interface#*

Example: `interface 1`

Last

Use the **last** command to display the last saved t-class statistics. The t-class statistics are displayed in the same format as they are for the **counters** command.

Syntax: last

Example: last

Last-Circuit-Class

Used only when monitoring Frame Relay.

Use the **last-circuit-class** command to display the last saved circuit class statistics. The c-class statistics are displayed in the same format as they are for the **counters-circuit-class** command.

Syntax: last-circuit-class

Example: last-circuit-class

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: exit

Chapter 12. Using and Configuring MAC Filtering

This chapter describes how to use medium access control (MAC) for specifying packet filters to be applied to packets during processing. It includes the following sections:

- "MAC Filtering and DLSw Traffic"
- "MAC Filtering Parameters" on page 12-2
- "Accessing the MAC Filtering Configuration Prompt" on page 12-3
- "MAC Filtering Configuration Commands" on page 12-4
- "Update Subcommands" on page 12-9

Filters are a set of rules applied to a packet to determine how the packet should be handled during bridging. MAC filtering affects only bridged traffic.

Note: MAC Filtering is allowed on tunnel traffic.

During the filtering process, packets are either processed, filtered, or tagged during bridging. The actions are:

- **Processed** – Packets are permitted to pass unaffected through the bridge.

- **Filtered** – Packets are not permitted to pass through the bridge.

- **Tagged** – Packets are allowed to pass through the bridge, but is marked with a number in the range of 1 through 64 based on a configurable parameter.

A MAC Filter consists of the following three objects:

1. Filter item – which is a single rule that is applied to the address field or an arbitrary window of data within a packet. The result of applying the rule is either a true (successful match) or false (no match) condition.

2. Filter list – which contains a list of one or more filter items.

3. Filter – which contains a set of filter-lists.

MAC Filtering and DLSw Traffic

You can filter incoming LLC traffic for the DLSw network by implementing MAC Filtering; you can filter incoming and outgoing DLSw traffic and you can forward incoming DLSw traffic to the bridge.

To set up a filter for LLC, use the *Bridge Net* number as the interface number for the filter. Determine the Bridge Net number by adding two to the number of interfaces configured for your router. Enter the **list devices** command at the Config> prompt, or enter **configuration** at the + prompt to see a list of interfaces.

In the following example, the Bridge Net number is 7.

```
Ifc 0 Token Ring                    Slot: 1  Port: 1
Ifc 1 Token Ring                    Slot: 1  Port: 2
Ifc 2 Token Ring                    Slot: 2  Port: 1
Ifc 3 Token Ring                    Slot: 2  Port: 2
Ifc 4 Ethernet                      Slot: 4  Port: 1
Ifc 5 Ethernet                      Slot: 4  Port: 2
```

When you set up a filter for the Bridge Net, for example, the router does not drop frames that match exclusive filters. Instead, it forwards those frames to the bridge.

MAC Filtering Parameters

You can specify some or all of the following parameters to create a filter:

- Source MAC address or destination MAC address
- Data to be matched within the packet
- Mask to be applied to the packet's fields to be filtered
- Interface number
- Input/Output designation
- Include/Exclude/Tag designation
- Tag value (if the tag designation is given)

Filter-Item Parameters

The following parameters are used to construct an address-filter-item:

- Address Type: SOURCE or DESTINATION
- Tag: a *tag-value*
- Address Mask: a *hex-mask*

Each filter-item specifies an address type (either SOURCE or DESTINATION) to match against the type in the packet.

The address mask is a string of numbers entered in hex, which is used in comparing the packet's addresses. The mask is applied to the SOURCE or DESTINATION MAC address of the packet before comparing it against the specified MAC address.

The address mask must be of equal length to the MAC address and specifies the bytes that are to be logically ANDed with the bytes in the MAC address before the equality comparison to the specified MAC address is made. If no mask is specified, it is assumed to be all 1s.

Filter-List Parameters

The following parameters are used to construct a filter-list:

- Name: an *ASCII-string*
- Filter-item list: *filter-item 1 . . . filter-item n*
- Action: INCLUDE, EXCLUDE, TAG(*n*)

A filter-list is built from one or more filter-items. Each filter-list is given a unique name.

Applying a filter-list to a packet consists of comparing each filter-item in the order in which the filter-items were added to the list. If any filter-item in the list returns a TRUE condition then the filter-list will return its designated action.

Filter Parameters

The following parameters are used to construct a filter:

- Filter-list names: *ASCII-string 1 . . . ASCII-string n*
- Interface number: an *IFC-number*
- Port direction: INPUT or OUTPUT
- Default action: INCLUDE, EXCLUDE, or TAG
- Default tag: a *tag-value*

A filter is constructed by associating a group of filter-list names with an interface number and assigning an INPUT or OUTPUT designation. The application of a filter to a packet means that each of the associated filter-lists should be applied to packets being received (INPUT) or sent (OUTPUT) on the specified numbered interface.

When a filter evaluates a packet to an INCLUDE condition, the packet is forwarded. When a filter evaluates a packet to an EXCLUDE condition, the packet is dropped. When a filter evaluates to a TAG condition, the packet being considered is forwarded with a tag.

An additional parameter of each filter is the default action, which is the result of non-match for all of its filter-lists. This default action is INCLUDE. It can be set to either INCLUDE, EXCLUDE, or TAG. In addition, if the default action is TAG, a tag value is also given.

Using MAC Filtering Tags

- MAC Address filtering is handled jointly by bandwidth reservation and the MAC Filtering feature (MCF) using tags. A user with bandwidth reservation is able to categorize bridge traffic, for example, by assigning a tag to it.

- The tagging process is done by creating a filter item in the MAC Filtering configuration console and then assigning a tag to it. This tag is then used to set up a bandwidth class for all packets associated with this tag. Tag values must currently be in the range 1 - 64.

  Note: Tag values are useful in only one respect with Version 1, Release 2. Version 1 Release 2 supports applying tags to *only* bridged packets and allows *only* the MAC address fields of the packet to be used in applying the tag.

- Once a tagged filter has been created in the MAC Filtering configuration process, the Bandwidth Reservation (BRS) **tag** configuration command is used to assign a BRS tag name (either TAG1, TAG2, TAG3, TAG4, or TAG5) to the MAC filter tag number. The BRS tag name is then used on the BRS **assign** configuration command to assign the corresponding MAC filter to a bandwidth traffic class and priority.

- Up to 5 tagged MAC addresses can be set from 1 to 5. TAG1 will be searched for first, then TAG2, all the way to TAG5.

Tags can also refer to "groups" in IP Tunnel. IP Tunnel end-points can belong to any number of groups, with packets assigned to a particular group through the tagging feature of MAC address filtering.

Accessing the MAC Filtering Configuration Prompt

Use the **feature** command from the CONFIG process to access the MAC filtering configuration commands. The **feature** command lets you access configuration commands for specific features outside the protocol and network interface configuration processes.

Enter a question mark after the **feature** command to obtain a listing of the features available for your software release. For example:

```
Config> feature ?
WRS
BRS
MCF
Feature name or number [MCF]?
```

To access the MAC filtering configuration prompt, enter the **feature** command followed by the *feature number* (3) or *short name* (MCF). For example:

```
Config> feature mcf
MAC Filtering user configuration
Filter config>
```

Once you access the MAC filtering configuration prompt, you can begin entering specific configuration commands. To return to the CONFIG prompt at any time, enter the **exit** command at the MAC filtering configuration prompt.

MAC Filtering Configuration Commands

This section summarizes and explains the MAC filtering configuration commands. Enter these commands at the Filter config> prompt.

Use the following commands to configure the MAC filtering feature.

Table 12-1. MAC Filtering Configuration Command Summary	
Command	**Function**
? (Help)	Displays all the MAC filtering commands or lists subcommand options for specific commands (if available).
Attach	Adds a filter list to a filter.
Create	Creates a filter list or an INPUT or OUTPUT filter.
Default	Sets the default action for the specified filter to EXCLUDE, INCLUDE, or TAG.
Delete	Removes all information associated with a filter list. Also deletes a filter that was created using the create filter command.
Detach	Deletes a filter list name from a filter.
Disable	Disables MAC Filtering entirely or to disables a particular filter.
Enable	Enables MAC Filtering entirely or enables a particular filter.
List	Lists a summary of all the filter lists and filters configured by the user. Also generates a list of attached filter lists for this filter and all subsequent information for the filter.
Move	Reorders the filter lists attached to a specified filter.
Reinit	Reinitializes the entire MAC Filtering system from an existing configuration without affecting the rest of the router.
Set-Cache	Changes the cache size for a filter.
Update	Adds or deletes information from a specific filter list. Brings you to a menu of appropriate subcommands.
Exit	Exits the MAC filtering configuration process.

? (Help)

Use the **?** (help) command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: ?

Attach

Use the **attach** command to add a filter list to a filter.

A filter is constructed by associating a group of filter lists with an interface number. A filter list is built from one or more filter items.

Syntax: at̲tach *filter-list-name filter-number*

Example: `attach atm_list 3`

Create

Use the **create** command to create a filter list or an INPUT or OUTPUT filter.

Syntax: create   list filter-list-name
                                filter input/output interface-number

list *filter-list-name*
    Creates a filter list. Lists are named by a unique string (Filter-list-name) of up to 16 characters of the user's choice. This name is used to identify a filter list that is being built. This name is also used with other commands associated with the filter list.

    Example: `create list newyork`

filter INPUT/OUTPUT *interface-number*
    Creates a filter and places it on the network associated with the INPUT or OUTPUT direction on the interface given by an interface number. By default this filter is created with no attached filter lists and has a default action of INCLUDE and is ENABLED.

    Example: `create filter INPUT 2`

Default

Use the **default** command to set the default action for the filter with a specified filter number to exclude, include, or tag.

Syntax: default   exclude *filter-number*
                                include *filter-number*
                                tag *tag-number filter-number*

exclude *filter-number*
    Sets the default action for the filter with a specified filter number to exclude.

    Example: `default exclude 3`

include *filter-number*
    Sets the default action for the filter with a specified filter number to include.

    Example: `default include 3`

tag *tag-number filter-number*
> Sets the default action for the filter with the specified filter number to TAG and sets the associated tag value to tag number.

> **Example:** `default tag 3 15`

Delete

Use the **delete** command to remove all information associated with a filter list and to free an assigned string as a name for a new filter list. If filter list is attached to a filter that has already been created by the user, then this command will display an error message on the console without deleting anything. In addition all filter items belonging to this list are also deleted

This command also deletes a filter that was created using the **create filter** command.

Syntax: delete   list   *filter-list*
                     filter   *filter-number*

list *filter-list*
> Removes all information associated with a filter-list and frees an assigned string as a name for a new filter-list. The filter-list must be a string entered by a previous **create list** command.

> If the filter-list is attached to a filter that has already been created by the user, then this command will display an error message on the console without deleting anything. All filter-items belonging to this list are also deleted when this command is used.

> **Example:** `delete list newyork`

filter *filter-number*
> Deletes a filter that was created using the **create filter** command.

> **Example:** `delete 3`

Detach

Use the **detach** command to delete a filter-list name (filter-list parameter) from a filter (filter-number parameter).

Syntax: detach   list   *filter-list*

Example: `detach list newyork`

Disable

Use the **disable** command to disable MAC Filtering entirely or to disable a particular filter.

Syntax: disable   all
                     filter   *filter-number*

all
> Disables MAC Filtering entirely. Filters are still set as ENABLED, however, if they were enabled previously.

> **Example:**   `disable all`

filter *filter-number*
> Disables a particular filter. The filter-number parameter corresponds to the
> numbers displayed in the **list filters** command.

> **Example:**　　`disable filter 3`

Enable

Use the **enable** command to enable MAC Filtering entirely or to enable a particular
filter.

Syntax: enable　　all
　　　　　　　　　　filter　*filter-number*

all
> Enables MAC Filtering entirely although filters themselves may still be set to
> DISABLED.

> **Example:**　　`enable all`

filter *filter-number*
> Enables a particular filter. The filter-number parameter corresponds to the
> numbers displayed in the **list filters** command.

> **Example:**　　`enable filter 3`

List

Use the **list** command to list a summary of all the filter-lists and filters configured by
the user. A list of all the filter-lists attached to a filter is not given. Other
information displayed includes:

- A list containing the state of the filtering system (ENABLE, DISABLE)
- The set of configured filter-list records
- Each of the configured filter records.

In addition, the following information is displayed for each filter:

- Filter number
- Interface number
- Filter direction (INPUT, OUTPUT)
- Filter state (ENABLE, DISABLE)
- Filter default action (TAG, INCLUDE, EXCLUDE).

This command also generates a list of attached filter-lists for this filter and all
subsequent information for the filter.

Syntax: list　　　all
　　　　　　　　　　filter　*filter-number*

Example:　　`list all`

filter *filter-number*
> Generates a list of attached filter-lists for the specified filter and all subsequent
> information for the filter.

> **Example:**　　`list filter 3`

Move

Use the **move** command to reorder the filter-lists attached to a specified filter (given by filter-number parameter). The list given by Filter-list-name1 is moved immediately before the list given by Filter-list-name2.

Syntax: move   *filter-list-name1 filter-list-name2 filter-number*

Example: `move newyork boston 13`

Reinit

Use the **reinit** command to reinitialize the entire MAC Filtering system from an existing configuration without affecting the rest of the router.

Syntax: reinit

Example: `reinit`

Set-Cache

Use the **set-cache** command to change the default cache size (16) to a number between 4 and 32768.

Syntax: set-cache   *cache-size filter-number*

Example: `set-cache 32 13`

Update

Use the **update** command to add information to or delete information from a specific filter-list. Using this command with the desired filter-list-name brings you to the `Filter filter-list-name Config>` prompt for that specific filter list. From this new prompt you can then change information in the specified list.

The new prompt level is used to add or delete filter-items from filter-lists. The order in which the filter-items are specified for a given filter-list is important as it determines the order in which the filter-items are applied to a packet.

Syntax: update *filter-list-name*

Example: `update newyork`

Exit

Use the **exit** command to return to the previous CONFIG prompt level.

Syntax: exit

Example: `exit`

Update Subcommands

This section summarizes and then explains the MAC filtering configuration subcommands. Enter these subcommands at the `Filter filter-list-name config>` prompt.

Table 12-2. Update Subcommands Summary	
Subcommand	**Function**
? (Help)	Displays all the update subcommands.
Add	Adds source or destination MAC address filters or a window filter. Adds filter-items to a filter-list.
Delete	Removes filter-items from a filter-list.
List	Lists a summary of all the filter-lists and filters configured by the user. Also generates a list of attached filter-lists for this filter and all subsequent information for the filter.
Move	Reorders the filter-lists attached to a specified filter.
Set-Action	Sets a filter-item to evaluate the INCLUDE, EXCLUDE or TAG (with a tag-number option) condition.
Exit	Exits the update subcommand configuration process.

Use the following subcommands to update a filter-list.

? (Help)

Use the **? (help)** subcommand to list the commands that are available from the current prompt level. You can also enter a ? after a specific subcommand name to list its options.

Syntax: ?

Example: ?

Add

Use the **add** subcommand to add filter-items to a filter-list. This subcommand specifically lets you add a hexadecimal number to compare against the source or destination MAC address. source or destination MAC address, or a sequence of window data with a mask to compare against packet data.

The order in which the filter-items are added to a given filter-list is important as it determines the order in which the filter-items are applied to a packet.

Each use of the **add** subcommand creates a filter-item within the filter-list. The first filter-item created is assigned filter-item-number 1, the next one is assigned number 2, etc. After a successful **add** subcommand is entered by the user, the router displays the number of the filter-item just added.

The first match that occurs stops the application of filter-items, and the filter-list evaluates to either INCLUDE, EXCLUDE or TAG, depending on the designated action of the filter-list. If none of the filter-items of a filter-list produce a match, then the default action (INCLUDE, EXCLUDE or TAG) of the filter is returned.

Syntax: a̲dd   source *hex-MAC-addr hex-Mask*

> destination *hex-MAC-addr hex-Mask*
> window MAC *offset-value hex-data hex-mask*
> window INFO *offset-value hex-data hex-mask*

source *hex-MAC-addr hex-Mask*
> Adds a hexadecimal number (with no 0x in front, a maximum of 16 numbers, and an even number of hex numbers) to compare against the source MAC address.
>
> The hex-mask parameter must be the same length as hex-MAC-address and is logically ANDed with the designated MAC address in the packet. The default hex-mask argument is to be all binary 1s.
>
> The hex-MAC-addr parameter can be specified in canonical or noncanonical bit order. A canonical bit order is specified as just a hex number (e.g., 000003001234). It may also be represented as a series of hex digits with a hyphen (-) between every two digits (e.g., 00-00-03-00-12-34).
>
> A noncanonical bit order is specified as a series of hex digits with a colon (:) between every two digits (e.g., 00:00:C9:09:66:49). MAC addresses of filter-items will always be displayed using either "-" or ":" to distinguish canonical from noncanonical representations.
>
> **Example:**   `add source 00-00-03-00-12-34  00-00-00-00-12-34`

destination *hex-MAC-addr hex-Mask*
> Acts identically to the add source subcommand, with the exception that the match is made against the destination rather than source MAC address of the packet.
>
> **Example:**   `add destination 00-00-03-00-12-34 00-00-00-00-12-34`

window MAC *offset-value hex-data hex-mask*
> Adds a sliding window filter-item using the specified offset (computed from the beginning of the frame) which matches the hex data with the mask against packet data.
>
> **Example:**   `add window mac 14 f4f403 ffffff`

window INFO *offset-value hex-data hex-mask*
> Similar to the **add window mac** command, except the offset is computed with respect to the beginning of the information field.
>
> **Example:**   `add window info 0 f4f403 ffffff`

Delete

Use the **delete** subcommand to remove filter-items from a filter-list. You delete filter items by specifying the filter-item-number assigned to the item when it was added.

When the **delete** subcommand is used, any gap created in the number sequence is filled in. For example, if filter-items 1, 2, 3, and 4 exist and filter-item 3 is deleted, then filter-item 4 will be renumbered to 3.

Syntax: <u>de</u>lete *filter-item-number*

Example: `delete 3`

List

Use the **list** subcommand to print out a listing of all the filter-item records represented in canonical and noncanonical form. the sliding window and MAC address filter-item records. The following information about each MAC-Address filter item is displayed:

- MAC address and address mask in canonical or noncanonical form.
- filter-item numbers
- address type (source or destination)
- filter-list action is given.

Syntax: list   canonical
                      noncanonical
                      mac-address canonical
                      mac-address noncanonical
                      window

canonical
> Prints out a listing of all the filter-item records within a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in canonical form and the address mask in canonical form. In addition the filter-list action is given.

mac-address canonical
> Prints out a listing of all the filter-item records within a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in canonical form and the address mask in canonical form. In addition the filter-list action is given.

> **Example:**   `list canonical`
>
>   `list mac-address canonical`

noncanonical
> Prints out a listing of all the filter-item records within a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in noncanonical form and the address mask in noncanonical form. In addition the filter-list action is given.

mac-address noncanonical
> Prints out a listing of all the filter-item records within a filter-list, giving the item numbers, the address type (SRC, DST), the MAC address in noncanonical form and the address mask in noncanonical form. In addition the filter-list action is given.

> **Example:**   `list noncanonical`
>
>   `list mac-address noncanonical`

window
> Prints out a listing of all the sliding window filter-item records within a filter-list, giving the item numbers, base, offset, data, and mask. In addition the filter-list action is given.

> **Example:**   `list window`

Move

The **move** subcommand reorders filter-items within the filter-list. The filter-item whose number is specified by *filter-item-name1* is moved and renumbered to be just before *filter-item-name2*.

Syntax: move   *filter-item-name1 filter-item-name2*

Example: `move 2 4`

Set-Action

The **set-action** subcommand lets you set a filter-item to evaluate the INCLUDE, EXCLUDE, or TAG (with a tag-number option) condition. If one of the filter-items of the filter-list matches the contents of the packet being considered for filtering, the filter-list will evaluate to the specified condition. The default setting is INCLUDE.

Syntax: set-action   *INCLUDE or EXCLUDE or TAG tag-number*

Example: `set action EXCLUDE`

Exit

Use the **exit** subcommand to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 13. Monitoring MAC Filtering

This chapter describes how to access the MAC Filtering console prompt and describes the available console commands. It includes the following sections:

- "Accessing the MAC Filtering Console Prompt"
- "MAC Filtering Console Commands"

Accessing the MAC Filtering Console Prompt

Use the **feature** command from the GWCON process to access the MAC filtering console commands. The **feature** command let you access console commands for specific router features outside of the protocol and network interface console processes.

Enter a question mark after the **feature** command to obtain a listing of the features available for your software release. For example:

```
+ feature ?
WRS
BRS
MCF
```

To access the MAC filtering console prompt, enter the **feature** command followed by the feature number (3) or short name (MCF). For example:

```
+ feature mcf
MAC Filtering user console
Filter>
```

Once you access the MAC filtering console prompt, you can begin entering specific console commands. To return to the GWCON prompt at any time, enter the **exit** command at the MAC Filtering console prompt.

MAC Filtering Console Commands

This section summarizes and explains the MAC filtering console commands. Enter these commands at the `Filter>` prompt.

Table 13-1. MAC Filtering Console Command Summary	
Command	**Function**
? (Help)	Displays all the MAC filtering commands or lists subcommand options for specific commands (if available).
Clear	Clears the "per filter" statistics listed in the list filter command.
Disable	Disables MAC Filtering globally or on a "per filter" basis.
Enable	Enables MAC Filtering globally or on a "per filter" basis.
List	Lists a summary of statistics and settings for each filter currently running in the router.
Reinit	Reinitializes the entire MAC Filtering system from an existing console without affecting the rest of the router.
Exit	Exits the MAC filtering console process.

Use the following commands to monitor the MAC filtering feature.

? (Help)

Use the ? (help) command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: `?`

Clear

Use the clear command to clear all the filter statistics.

Syntax: clear   all
                   filter   *filter-number*

all
     Clears the statistics listed by the **list all** command.

     Example: `clear all`

filter *filter-number*
     Clears the statistics listed by the **list filter** command.

     Example: `clear filter 3`

Disable

Use the **disable** command to disable MAC filtering globally. This command does not individually disable each filter.

The command also disables a filter as specified by filter-number. This filter is disabled without modifying configuration records. If no argument is given, MAC filtering is globally disabled.

Syntax: disable   all
                    filter   *filter-number*

all
     Disables MAC filtering globally. This command does not individually disable each filter.

     Example: `disable all`

filter filter-number
     Disables the filter that is specified by the filter number. This filter is disabled without modifying configuration records. If no filter number is given, MAC filtering is globally disabled.

     Example: `disable filter 3`

Enable

Use the **enable** command to enable MAC filtering globally. This command does not individually enable each filter.

The command also enables a filter as specified by filter-number. This filter is enabled without modifying configuration records. If no argument is given, MAC filtering is globally enabled.

Syntax: enable   all
                filter *filter-number*

all
> Enables MAC filtering globally. This command does not individually enable each filter.

> **Example:**   `enable all`

filter *filter-number*
> Enables the filter that is specified by the filter number. This filter is enabled without modifying configuration records. If no filter number is given, MAC filtering is globally enabled.

> **Example:**   `enable filter 3`

List

Use the **list** command to list a summary of statistics and settings for each filter currently running in the router. The following information is displayed for each filter when the **list all** command is used:

- default action
- cache size
- default tag
- state (enabled/disabled)
- number of packets which have been filtered as INCLUDE, EXCLUDE or TAG.

In addition, the following information is also displayed by the **list filter** command for a specified filter:

- all information displayed by the list all command

- all the filter-lists currently running in this filter including:

  - list name
  - list action
  - list tag
  - number of packets which have been filtered by each filter-list.

Syntax: list   all
                filter *filter-number*

all
> Lists statistics and settings for each filter currently running in the router.

> **Example:**   `list all`

filter *filter-number*
> Generates statistics and settings for each filter plus all the filter-lists currently running in this filter.

> **Example:**   `list filter 3`

Reinit

Use the **reinit** command to reinitialize the entire MAC Filtering system from an existing configuration without affecting the rest of the router.

Syntax: reinit

Example: `reinit`

Exit

Use the **exit** command to return to the previous GWCON prompt level.

Syntax: exit

Example: `exit`

Chapter 14. Configuring WAN Restoral

This chapter includes the following sections:

- "Before You Begin"
- "Configuration Procedure"
- "Secondary Dial Circuit Configuration" on page 14-2
- "WAN-Restoral Configuration Commands" on page 14-3.

Before You Begin

Before you configure WAN-Restoral, you must have the following:

1. A primary serial interface (leased line) configured on the router. You can use any serial interface on the router.

2. A secondary interface with the associated dial circuits configured on the router. You can use an ISDN interface or a V.25 bis interface as the secondary link.

3. A secondary dial circuit configured to dial when the primary interface goes down. To configure a dial circuit to do this, set the idle timer to zero using the **set idle** command at that dial circuit's `Circuit Config>` prompt.

4. A secondary dial circuit at one end of the link configured to receive calls only. (Use the **set calls inbound** command at the `Circuit Config>` prompt.)

5. A secondary dial circuit at the other end of the link configured to send calls only. (Use the **set calls outbound** command at the `Circuit Config>` prompt.)

   Note: Do not configure any protocol addresses on the secondary interface or the dial circuit. The protocol assignments for the primary interface are used on the secondary link (dial circuit) when it is active.

Data-Link Layer Configuration

The primary and secondary links must be configured for the same data-link layer protocol (that is, PPP).

All of the upper layer PPP configuration should be exactly the same between the primary link and the secondary link.

For information on PPP, V.25 bis, or ISDN, see the appropriate chapters in this guide.

Configuration Procedure

This section describes the steps required to configure WAN-Restoral. Before you begin, use the **list device** command at the `Config>` prompt to list the interface numbers of different devices.

Follow these steps to configure WAN-Restoral on the router:

1. Display the `WRS Config>` prompt by entering the **feature wrs** command at the `Config>` prompt. For example:

   ```
   Config>feature wrs
   WAN Restoral user configuration
   WRS Config>
   ```

2. Assign a secondary dial circuit to the primary interface. This dial circuit will back up the primary interface. For example:

   ```
   WRS Config>add secondary-circuit
   Secondary interface number  [0]? 3
   Primary interface number  [0]? 1
   ```

3. Enable WAN-Restoral on the secondary dial circuit that you added. For example:

   ```
   WRS Config>enable secondary-circuit
   Secondary interface number  [0]? 3
   ```

4. Globally enable WAN-Restoral on the router. For example:

   ```
   WRS Config>enable wrs
   ```

5. Restart the router for configuration changes to take effect.

Secondary Dial Circuit Configuration

To configure a dial circuit:

1. Determine the dial-circuit interface number: To do this, type:

   ```
   Config> list device
   ```

   If no dial-circuit interface is listed, add a dial-circuit interface by typing:

   ```
   Config> add device dial-circuit

   Adding device as interface 3
   Defaulting Data-link protocol to PPP
   Use "net 3" command to configure circuit parameters
   ```

2. Configure the secondary interface (dial circuit) to have the same data-link type as the primary interface (PPP) from the Config> prompt as follows:

   ```
   Config> set data PPP
   Interface Number [0]? 3
   ```

3. Access the dial circuit configuration prompt (Circuit Config>) by entering **network** *interface#*.

   ```
   Config> network 3
   ```

4. Set the dial circuit idle timer to 0 (0=active) as follows:

   ```
   Circuit Config> set idle 0
   ```

5. Set one end of the backup connection to receive calls (for example, router A) as follows:

   ```
   Circuit Config> set calls inbound
   ```

6. Set the other end of the backup connection to initiate calls (for example, router B) as follows:

   ```
   Circuit Config> set calls outbound
   ```

Notes:

1. Do not use the **set calls both** command. Setting these individually will help prevent the collisions of incoming and outgoing connection attempts.

2. Do not configure any forwarder (for example, IP, IPX, etc.) addresses on the V.25 bis interface or the dial circuit. The protocol assignments for the primary interface are used on the secondary interface (dial circuit) when it is active.

3. For V.25 bis configuration instructions, refer to Chapter 35, "Using and Configuring the V.25 bis Network Interface" on page 35-1.

WAN-Restoral Configuration Commands

The WAN-Restoral configuration commands allow you to create or modify the WAN Restoral interface configuration. This section summarizes and explains the WAN Restoral configuration commands.

Table 14-1 lists the WAN Restoral configuration commands and their function. Enter these commands at the WRS Config> prompt. To access WRS Config>, enter **feature wrs** at the Config> prompt.

Table 14-1. WAN Restoral Configuration Commands Summary	
Command	**Function**
? (Help)	Lists the configuration commands or lists any parameters associated with that command.
Add	Adds a mapping of primary to secondary backup.
Disable	Disables WRS, or an individual secondary-circuit mapping.
Enable	Enables WRS, or an individual secondary-circuit mapping.
List	Displays the current Restoral configuration.
Remove	Removes a primary to secondary mapping created by add.
Exit	Exits the WAN Restoral configuration process.

? (Help)

Use the **?** (help) command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
Add
Disable
Enable
List
Remove
Exit
```

Add

Use the **add** command to identify a secondary backup dial circuit for a particular primary serial link.

Note: When configuring BRS over a primary link and WAN Restoral on a secondary link such as ISDN, configure the following:

On the dial-circuit link, configure BRS. Typically when WAN Restoral is configured on the backup link, WRS takes the identity of the primary link. This is not true for BRS; therefore, BRS needs to be configured on both the primary link and the secondary link.

Syntax: <u>a</u>dd   <u>s</u>econdary-circuit . . .

secondary-circuit

The **add secondary-circuit** command binds a secondary interface to a primary interface. Both interfaces must have previously been configured. You can only assign one secondary interface to a primary and vice-versa.

Example: **add secondary-circuit**

```
WRS Config>add secondary-circuit
Secondary interface number  [0]? 4
Primary interface number  [0]? 1
```

Secondary interface number	This is the dial circuit interface number previously assigned to the secondary interface when the device was added. Any dial circuit is an eligible secondary interface. The default is 0.
Primary interface number	This is the interface number of the primary interface previously assigned when the device was added. A primary interface can be any previously defined leased-line running PPP. The default is 0.

Disable

Use the **disable** command to disable the WAN Restoral function or to disable the restoral of a particular primary interface by its associated secondary interface.

Syntax: d̲isable   s̲econdary-circuit . . .
                       w̲rs

secondary-circuit *interface#*

Disables the restoral of a particular primary interface by its associated secondary interface until the next **enable secondary** command at the WRS console. Both interfaces must have been previously configured and bound together in the WRS configuration.

Example: **disable secondary-circuit**

```
WRS Config> disable secondary-circuit
Secondary interface number  [0]? 3
```

Secondary interface number	This is the number of the secondary interface previously configured with the **add secondary** command. The default is 0.

wrs

Disables the WAN Restoral feature globally on the router.

Example: disable wrs

```
WRS Config> disable wrs
```

Enable

Use the **enable** command to enable the WAN Restoral interface function or enable the restoral of a particular primary interface by its associated secondary interface.

Syntax: e̲nable   s̲econdary-circuit . . .
                       w̲rs

secondary-circuit *interface#*

Enables the restoral of a primary link by the indicated secondary link.

Example: **`enable secondary-circuit`**

```
WRS Config>enable secondary-circuit
Secondary interface number  [0]? 3
```

Secondary interface   This is the number of the secondary interface previously
number   configured with the **add secondary** command. The default is
                          0.

wrs

Enables the function of the WAN Restoral feature on the router.

Example: **`enable wrs`**

```
WRS Config> enable wrs
```

List

Use the **list** command to display configuration information on restored circuits. This
command allows the console operator to query statistical information about each
configured primary-to-secondary circuit binding.

Syntax: l̲ist

Example: **`list`**

```
WAN Restoral is enabled.

                                                Secondary
Primary Interface          Secondary Interface  Enabled
-----------------          -------------------  --------
1 - WAN PPP                3 - PPP Dial Circuit    Yes
```

Primary Interface   This is the interface number and underlying device type.

Secondary Interface   This is the interface number and underlying device type.

Secondary Enabled   Indicates whether or not the restoral mapping is enabled. Enabled
                          can be either Yes or No. The default is No. (Note that when the
                          secondary is initially added, the restoral is not enabled.)

Remove

Use the **remove** command to delete the mapping of a secondary (backup) interface
to the primary interface.

Syntax: r̲emove   s̲econdary-circuit . . .

secondary-circuit

Removes the mapping of a secondary (backup) interface to the primary
interface. Both interfaces must have been previously assigned and bound
together using the **add secondary-circuit** command.

Example: **`remove secondary-circuit`**

```
WRS Config> remove secondary-circuit
Secondary interface number  [0]? 3
Primary interface number  [0]? 1
```

Secondary interface   This is the number of the secondary interface previously
number   configured with the **add secondary** command. The default is
                          0.

Primary interface   This is the interface number of the primary interface previously
number   bound to the secondary being removed. The default is 0.

Exit

Use the **exit** command to return to the Config> prompt.

Syntax: <u>ex</u>it

Example: exit

Chapter 15. Monitoring WAN-Restoral

This chapter describes the WAN-restoral monitoring commands. It includes the following sections:

- "Accessing the WAN-Restoral Interface Console Process"
- "WAN-Restoral Monitoring Commands"

Accessing the WAN-Restoral Interface Console Process

To access the WAN-Restoral interface console process, enter the following command at the GWCON (+) prompt:

```
+ feature wrs
```

WAN-Restoral Monitoring Commands

The WAN-Restoral (WRS) console commands allow you to monitor the state of the secondary and primary interfaces protected by WRS. Any modifications to the operational state of WRS made through the console interface are not maintained across restarts.

Access the WRS prompt by entering **feature wrs** at the GWCON (+) prompt. Table 15-1 lists the WRS commands and their functions, and the following sections explain the commands.

Table 15-1. WAN-Restoral Monitoring Commands	
Command	**Function**
? (Help)	Lists the monitoring commands or lists any parameters associated with that command.
Clear	Clears the monitoring statistics displayed using the **list** command.
Disable	Disables the WRS, or an individual secondary.
Enable	Enables the WRS, or an individual secondary.
List	Displays the monitoring information on one or all restored circuits.
Exit	Returns to the GWCON (+) prompt level.

? (Help)

Use the **?** (help) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

```
Clear
Disable
Enable
List
Exit
```

Clear

Use the **clear** command to clear WAN-Restoral statistics that are displayed using the **list** command.

Syntax: c̲lear

Example: **clear**

> WRS> **clear**

Note: This command clears *Longest restoral period*, but does not clear the *Most recent restoral period*. For the screen display, refer to the example in the **list** command.

Disable

Use the **disable** command to disable the WAN-Restoral feature completely, or disable the restoral of a particular primary interface by its associated secondary interface.

Syntax: d̲isable   s̲econdary-circuit
>   w̲rs

secondary-circuit
> Disables the restoral of a particular primary interface by its associated secondary interface until the next **restart**, **reload**, or **enable secondary** command. Both interfaces must have been previously configured and bound together in the WRS configuration.
>
> **Note:** Normally, in task 5 (console), the **disable** command causes the interface to be inactive and stay inactive. For WAN-Restoral secondary, however, this is not the case. The **disable** command applied to the secondary interface does not disable the interface itself. It disables only the current call (that is, causes any active call to be disconnected.) To disable use of the secondary circuit, you need to **disable secondary** at the WAN-Restoral menu and also in the console process.
>
> **Example:** **disable secondary-circuit**
>
> > WRS>**disable secondary-circuit**
> > Secondary interface number [0]? **3**
>
> *Secondary interface number*   This is the number of the secondary interface previously configured with the **add secondary** command. The default is 0.

wrs
> Disables WAN-Restoral on the router until the next **restart**, **reload**, or **enable WRS** command.
>
> **Example:** **disable wrs**
>
> > WRS> **disable wrs**

Enable

Use the **enable** command to enable the WAN-Restoral interface or enable the restoral of a primary link by a secondary circuit.

Syntax: <u>en</u>able   <u>s</u>econdary-circuit
                    <u>w</u>rs

secondary-circuit
    Enables the restoral of a primary link by the indicated secondary link.

   Example: `enable secondary-circuit`

           WRS> **`enable secondary-circuit`**
           Secondary interface number [0]? **3**

   Secondary interface   This is the number of the secondary interface previously
   number   configured with the **add secondary** command. The default is
                             0.

wrs
    Enables the function of the WAN-Restoral feature on the router.

   Example: `enable wrs`

           WRS> **`enable wrs`**

List

Use the **list** command to display monitoring information on one or all restored circuits.

Syntax: <u>l</u>ist   <u>a</u>ll
                    <u>se</u>condary-circuit
                    <u>su</u>mmary

all
    Provides summary information, followed by the specific information, for each secondary interface.

   Example: `list all`

           WAN Restoral is enabled with 1 circuit(s) configured

           Total restoral attempts   =   3   completions   =   2
           Total packets forwarded   =   346

           Longest restoral period in hrs:min:sec   00:08:20

                                                              Secondary
           Primary Interface   Secondary Interface   Enabled
           -------------------   -------------------   --------
           1 PPP/0 Point to Poi   3 PPP/1 Point to Poi   Yes

           Router primary interface state = Up
           Router secondary interface state = Available
           Restoral Statistics:

             Primary restoral attempts = 6   completions = 5
             Restoral packets forwarded = 346
             Most recent restoral period in hrs:min:sec   00:08:20

Total restoral attempts	The number of times the primary link failed, causing the router to try to bring up a secondary link.
Completions	The number of successful restoral attempts when the secondary link came up and was used.
Total packets forwarded	The total number of packets forwarded across the secondary interface. It is the sum of both directions, and is cumulative over all successful restores, until the restart or clear restoral-statistics command is issued.
Longest restoral period	This field displays in hours, minutes, and seconds the longest amount of time a restoral was in operation, not counting any current usage.
Primary Interface	The interface that is being backed up by its associated secondary interface.
Secondary Interface	The dial circuit that is being used to back up the associated primary interface.
Secondary Enabled	Indicates that restoral of this primary interface is currently enabled.
Router primary interface state	Indicates that the primary interface is in one of the following states:

       Up - Indicates that the link is up.

       Down - Indicates that the link is down.

       Disabled - Indicates that the operator has disabled the link.

       Not present - Indicates that the link is configured but there is a hardware problem.

Router secondary interface state	Indicates that the associated secondary interface is in one of the following states:

       Up - Indicates that the link is up.

       Down - Indicates that the link is down. This also occurs when the base network for the secondary interface is disabled either at the `Config`> prompt or at the operator console.

       Available - Indicates that the link is in the waiting mode.

Restoral Statistics:	
Primary Restoral Attempts	The number of times the primary failed, causing the router to try to bring up this secondary link.
Restoral Packets forwarded	This field indicates the total number of packets forwarded.
Most Recent Restoral Period	This indicates how long the secondary was up, the last time it was used, or currently in use.

secondary-circuit
    Provides totals for each secondary circuit. Allows the console operator to retrieve the WAN-Restoral state and associated statistics for each secondary interface and its associated primary mapping.

    Example: `list secondary-circuit`

```
Secondary interface number  [0]? 1

                                          Secondary
Primary Interface          Secondary Interface    Enabled
-------------------        --------------------   ---------
1  PPP/0  Point to Poi     3  PPP/1  Point to Poi     Yes

Router primary interface state = Up
Router secondary interface state = Available
Restoral Statistics:

  Primary restoral attempts =        6   completions =    5
  Restoral packets forwarded =     346
    Most recent restoral period in hrs:min:sec        00:08:20
```

Primary Interface	The interface that is being backed up by this associated secondary interface.
Secondary Interface	The dial circuit that is being used to back up the associated primary interface.
Secondary Enabled	Indicates whether restoral of this primary interface is currently enabled.
Router Primary Interface State	Indicates that the primary interface state is one of the following:
	Up - Indicates that the link is up.
	Down - Indicates that the link is down.
	Disabled - Indicates that the operator has disabled the link.
	Not present - Indicates that the link is configured but there is a hardware problem.
Router Secondary Interface State	Indicates that the associated secondary interface state is one of the following:
	Up - Indicates that the link is up.
	Down - Indicates that the link is down. This also occurs when the base network for the secondary is disabled either at the Config> prompt or at the operator console.
	Available - Indicates that the link is in the waiting mode.
	Testing - Indicates that the link is in the process of establishing a connection.
Restoral Statistics:	
Primary Restoral Attempts	The number of times the primary failed, causing the router to try to bring up a secondary link.
Restoral Packets forwarded	This field indicates the total number of packets forwarded.
Most Recent Restoral Period	This indicates how long the secondary was up, the last time it was used or during the current restoral use.

summary
Provides totals for each secondary circuit.

Example: `list summary`

```
                       WAN Restoral is enabled with 3 circuit(s) configured

                  Total restoral attempts =     3 completions =     2
                  Total packets forwarded =   346
                  Longest restoral period in hrs:min:sec   00:08:20

                 Primary Interface and State     Secondary Interface and State
                 --------------------------     ----------------------------------
                  1  PPP/0  - Up                  3  PPP/1 - Available
```

Total restoral attempts	The number of times the primary failed, causing the router to try to bring up a secondary link.
Completions	The number of successful restoral attempts when the secondary came up and was used.
Total packets forwarded	The total number of packets forwarded across the secondary interface. It is the sum of both directions, and is cumulative over all restoral periods until the restart or clear restoral-statistics command is used.
Longest restoral period	This field displays in hours, minutes, seconds the longest amount of time restoral was in use, not counting the current usage.
Primary Interface and State	The interface that is being backed up by its associated secondary. Valid states are:
	Up - Indicates that the link is up.
	Down - Indicates that the link is down.
	Disabled - Indicates that the operator has disabled the link.
	Not present - Indicates that the link is configured but there is a hardware problem.
Secondary Interface and State	The dial circuit that is being used to back up the associated primary. Valid states are:
	Up - Indicates that the link is up.
	Down - Indicates that the link is down. This also occurs when the base network for the secondary is disabled either at the `Config>` prompt or at the operator console.
	Testing - Indicates that the link is in the process of establishing a connection.
	Available - Indicates that the link is in the waiting mode.

Exit

Use the **exit** command to return to the GWCON (+) prompt.

Syntax: <u>exi</u>t

Example: exit

Chapter 16. Getting Started with Network Interfaces

The chapters of this book describe how to configure and monitor network interfaces and link layer protocols supported by the Router. The purpose of this chapter is to give you some basic configuration and monitoring guidelines. This chapter also provides you with basic procedures and information needed for monitoring the interfaces via the GWCON **interface** command. Sections in this chapter include:

* "Before You Continue"
* "Network Interfaces and the GWCON Interface Command"
* "Accessing Network Interface Configuration and Console Processes"
* "Accessing Link Layer Protocol Configuration and Console Processes"

Before You Continue

Before you continue, make sure that you have familiarized yourself with the procedures necessary for accessing the network interface configuration processes.

For more information on these procedures, refer to the sections that follow in this chapter.

Network Interfaces and the GWCON Interface Command

When configuring network interfaces, you may find it necessary to display certain information about specific interfaces. While some interfaces have their own console processes for monitoring purposes, the router displays statistics for *all* installed network interfaces when you use the **interface** command from the GWCON environment. (Refer to "Interface" on page 6-9.)

Accessing Network Interface Configuration and Console Processes

The follow references contain the background information and examples of how to access the configuration and console prompts for interfaces.

Refer to "Accessing Network Interface Configuration and Console Processes" on page 1-15, "Accessing the Network Interface Configuration Process" on page 1-16, and "Accessing the Network Interface Console Process" on page 1-19 for complete information on accessing interface configuration and console processes. Accessing these processes allows you to change and monitor software configurable parameters for network interfaces used in your router.

Accessing Link Layer Protocol Configuration and Console Processes

Refer to Chapter 1, "Getting Started (Introduction to the User Interface)" for complete information on accessing the protocol configuration and console processes. Accessing these processes allows you to change and monitor configurable parameters for Link Layer protocols supported by your router.

Chapter 17. Configuring IEEE 802.5 Token-Ring Network Interfaces

This chapter describes how to set software configurable information for token-ring interfaces in the router. It includes the following sections:

- "Accessing the Interface Configuration Process"
- "Token-Ring Configuration Commands"

Accessing the Interface Configuration Process

To display the `TKR config>` prompt, enter the network command followed by the interface number of the Token-Ring interface. For example:

```
Config>network 0
Token-Ring interface configuration
TKR Config>
```

Use the **list devices** command at the `Config>` prompt to display a list of interface numbers configured on the router.

Note: Whenever you change a parameter, you must restart the router for the changes to take effect.

Token-Ring Configuration Commands

This section summarizes and then explains the token-ring configuration commands. Enter the commands at the `TKR config>` prompt. Table 17-1 lists token-ring configuration commands.

Table 17-1. Token-Ring Configuration Command Summary

Command	Function
? (Help)	Displays all the token-ring commands or lists subcommand options for specific commands.
Frame	Sets the NetWare IPX encapsulation type.
List	Displays the selected token-ring interface configuration.
LLC	Accesses the LLC configuration environment and subcommands.
Media	Sets the media-type as shielded or unshielded.
Packet-size	Changes packet-size defaults for all token-ring networks.
Set	Sets the aging timer for the RIF cache and the physical (MAC) address.
Source-routing	Enables or disables source-routing on the interface.
Speed	Sets the interface speed in Mbps.
Exit	Exits the token-ring configuration process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
EXIT
FRAME
LIST
LLC
MEDIA
PACKET-SIZE
SET
SOURCE-ROUTING
SPEED Mb/sec
```

media ?

Frame

Use the **frame** command to set the NetWare IPX encapsulation type. Enter one of the following:

Option	Description	Syntax
Token-Ring using MSB	Uses the standard 802.2 IPX header with the noncanonical Token-Ring address bit ordering (MSB).	**frame token-ring msb**
Token-Ring using LSB	Uses the 802.2 IPX header with the canonical address bit ordering (LSB).	**frame token-ring lsb**
Token-Ring with 802.2 SNAP using MSB	Uses the 802.2 format with a SNAP header and noncanonical address bit ordering. This encapsulation is used primarily in bridging environments.	**frame token-ring_snap msb**
Token-Ring with 802.2 SNAP using LSB	Uses the 802.2 format with a SNAP header and canonical address bit ordering.	**frame token-ring_snap lsb**

Syntax: frame   *encapsulation type*

Example: **frame token_ring msb**

> **Note:** The **frame** command cannot be used in a network configuration process to set an encapsulation until the interface has been properly configured via the IPX configuration process.

List

Use the **list** command to display the current configuration for the token-ring interface.

Note: If the MAC address is 0, the default station address is used.

Syntax: <u>l</u>ist

Example: `list`

```
Token-Ring configuration:

    Packet size (INFO field): 2052
Speed:                    16 Mb/sec
Media:                    Shielded

RIF Aging Timer:          120
Source Routing:           Enabled
MAC Address:              000000000000
NetWare IPX encapsulation: TOKEN-RING MSB
```

Packet size	Size of the token-ring packet.
Speed	Speed of the network.
Media	Type of media the network uses, shielded or unshielded.
RIF Aging Timer	Amount of time that the router holds the information contained in the Routing Information Field (RIF).
Source Routing	Status of the source-routing feature, enabled or disabled.
MAC Address	Configured MAC address that was set with the **set physical-address** command. If all zeros are displayed, the MAC address is the default address.
Netware IPX encapsulation	Configured NetWare IPX encapsulation type that was set with the **frame** command.

LLC

Use the **LLC** command to access the LLC configuration environment. See "LLC Configuration Commands" on page 19-1 for an explanation of each of these commands.

Syntax: <u>llc</u>

Example: `llc`

> `LLC config>`

Note: If APPN is not included in your router software load, you will receive the following message if you try to use this command:

> `LLC configuration is not available for this network.`

The LLC configuration environment is only available if APPN is included in the software load.

Media

Use the **media** command to change the network media type. The default media type is STP cable. Valid media type values are shielded and unshielded. Enter the media command followed by the *media-type*.

Syntax: <u>m</u>edia *media-type*

Example: `media unshielded`

Packet-Size

Use the **packet-size** command to change packet-size defaults for all token-ring networks. Enter the **packet-size** command followed by the desired number of bytes.

Table 17-2. Token-Ring 4/16 Valid Packet Sizes	
Network Data Speed	**Values (# of bytes)**
4 Mbps	2052, 4399
	Note: If you configure a packet size greater than 4399, the software sets the packet size to 4399.
16 Mbps	1470, 2052, 4399, 8130, 11407, and 17749

Note: If packet sizes are increased, buffer memory requirements will also increase.

Syntax: packet-size *#bytes*

Example: `packet-size 4399`

Set

Use the **set** command to set the Routing Information Field (RIF) timer and the physical (MAC) address.

Syntax: set   physical-address
                  rif-timer

physical-address
>   Indicates whether you want to define a locally administered address for the Token-Ring interface's MAC sublayer address, or use the default factory station address (indicated by all zeroes). The MAC sublayer address is the address that the Token-Ring interface uses to receive and transmit frames.
>
>   **Note:** Pressing **Return** leaves the value the same. Entering **0** and pressing **Return** causes the router to use the factory station address. The default is to use the factory station address.
>
>   **Example:** `set physical-address`
>
>   `MAC address in 00:00:00:00:00:00 form []?`

rif-timer
>   Sets the maximum amount of time (in seconds) that the information in the RIF is maintained before it is refreshed. The default is 120.
>
>   **Example:** `set rif-timer`
>
>   `RIF aging timer value [120]?  120`

Source-routing

Use the **source-routing** command to enable or disable end station source routing. Source routing is the process by which end stations determine the source route to use to cross source routing bridges. Source routing allows the IP, IPX, and AppleTalk Phase 2 protocols to reach nodes on the other side of the source routing bridge.

This switch is completely independent of whether this interface is providing source routing via the SRT forwarder. The default setting is enabled.

Some stations cannot properly receive frames with a Source Routing RIF on them. This is especially common among NetWare drivers. Disabling source routing in this situation will allow you to communicate with these stations.

Source routing should be enabled only if there are source-routing bridges on this ring that you want to bridge IP, IPX, and AppleTalk Phase 2 packets through. Source routing must also be enabled so LLC test response messages can be returned.

Syntax: <u>so</u>urce-routing   <u>e</u>nable
<u>d</u>isable

Example: `source-routing enable`

Speed

Use the **speed** command to change data speed. The default speed is 4 Mbps. Enter the **speed** command followed by the speed-value (in Mbps).

Syntax: <u>sp</u>eed *speed-value*

Example: `speed 16`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: <u>e</u>xit

Example: `exit`

Chapter 18. Monitoring IEEE 802.5 Token-Ring Network Interfaces

This chapter describes how to monitor specific Token-Ring interfaces in the router by using either the interface console commands or the GWCON interface command. It includes the following sections:

- "Accessing the Interface Console Process"
- "Token-Ring Interface Console Commands"
- "Token-Ring Interfaces and the GWCON Interface Command" on page 18-3

Accessing the Interface Console Process

To display the token-ring monitoring prompt (TKR>), enter the network command followed by the interface number of the Token-Ring interface. For example:

```
+network 0
TKR>
```

Use the **list devices** command at the Config> prompt to display a list of interface numbers configured on the router.

Follow the procedure described in Chapter 16, "Getting Started with Network Interfaces" on page 16-1 to access the interface console process for the interface described in this chapter. Once you have accessed the desired interface console process, you can begin entering console commands.

Token-Ring Interface Console Commands

This section summarizes and explains the Token-Ring console commands. Enter commands at the TKR> monitoring prompt. Table 18-1 lists the console commands.

Table 18-1. Token-Ring Console Command Summary	
Command	**Function**
? (Help)	Displays all the Token-Ring commands or lists subcommand options for specific commands.
Dump	Displays a dump of the RIF cache.
LLC	Displays the LLC console monitoring prompt.
Exit	Exits the Token-Ring console process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: ?

```
dump
llc
exit
```

Dump

When source routing is enabled in the `tkr config>` process, you can use the **dump** command to request a dump of the RIF cache contents.

Syntax: <u>d</u>ump

Example: **dump**

```
MAC address    State    Usage   RIF
0000C90B1A57   ON_RING  Yes     0220
```

MAC address   Displays the MAC address of the Token-Ring interface.

State   Displays one of the interface states:

> On_ring - indicates that a RIF was found for a node on the ring.

> Have_route - indicates that a RIF was found for a node on a remote ring.

> No_route - is displayed for a brief period of time as an explorer frame is sent out and the router is waiting for a return.

> Discovering - indicates that the router sent an explorer frame to rediscover the RIF.

> St_route - indicates that a route obtained from a Spanning tree explorer.

Usage   Indicates that a RIF was used in a packet. The number is arbitrary and has no functional significance.

RIF   Displays a code that indicates the RIF in hexadecimal.

> **Note:** The RIF is displayed only if Source Route Bridging is enabled on the token-ring interface.

> - NetBIOS RIF data can be displayed using the following sequence of commands: **talk 5, protocol ASRT, name-caching, list cache rifs.**

> - Data Link Switching RIF data can be displayed using the following sequence of commands: **talk 5, protocol dlsw, list llc2 session all.**

LLC

Use the **LLC** command to access the LLC monitoring prompt. LLC commands are entered at this new prompt. See "LLC Monitoring Commands" on page 20-1 for an explanation of each of these commands.

Syntax: <u>ll</u>c

Example: **llc**

```
LLC user monitoring
LLC>
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: <u>e</u>xit

Example: **exit**

Token-Ring Interfaces and the GWCON Interface Command

While Token-Ring interfaces have their own console processes for monitoring purposes, the router also displays complete statistics for installed network interfaces when you use the **interface** command from the GWCON environment.

Statistics Displayed for 802.5 Token-Ring Interfaces

The following statistics display when you enter the **interface *<net #>*** command for a Token-Ring interface from the GWCON environment.

```
                                    Self-Test  Self-Test  Maintenance
Nt Nt'  Interface  Slot-Port        Passed     Failed     Failed
0  0    TKR/0      Slot: 1 Port: 1     1          0          0

  Token-Ring/802.5 MAC/data-link on Token-Ring interface

  Physical address       08005AFE0106
  Microcode Level        ww19cg
  Network speed          16 Mbps
  Max packet size (INFO) 2052
  Handler state          Ring open

  # times Signal lost     0    # times Beaconing      0
  Hard errors             0    Lobe wire faults       0
  Auto-removal errors     0    Removes received       0
  Ring recovery actions   0

  Line errors             0    Burst errors           0
  ARI/FCI errors          0    Inputs dropped         0
  Frame copy errors       0    Token errors           0
  Lost frames             0
  Input overflows         0    Driver output errors   0
```

The following section describes general interface statistics:

Nt	Global interface number
Nt'	Applies only to dial circuits
Interface	Interface name and Number of this interface within interfaces of type "intrfc"
Port	Port number
Slot	Slot number
Self-Test: Pass	Number of times self-test succeeded
Self-Test: Fail	Number of times self-test failed
Maint: Fail	Number of maintenance failures

The following section describes the statistics displayed that are specific to the Token-Ring interfaces:

Physical address	Specifies the physical address of the Token-Ring interface.
Network speed	Specifies the speed of the Token-Ring network that connects to the interface. The Network Speed counter displays the number of packets that the interface can pass per second.

Max packet size (info)	Displays the maximum packet size configured for that interface. The Max Packet Size counter displays the maximum length, in bytes, of a packet that the interface transmits or receives. This counter is user-defined.
Handler state	Displays the current state of the Token-Ring handler. The Handler state counter displays the state of the handler after the self-test runs.
Ring status	Last Ring Status of the Token Ring interface.

SIGL SIGNAL_LOSS   The interface has detected a loss of signal on the ring.

HERR HARD_ERROR   The interface is presently transmitting or receiving beacon frames on the ring.

SERR SOFT_ERROR   The interface has transmitted a report error MAC frame.

BEAC TRANSMIT_BEACON   The interface is transmitting beacon frames to or from the ring.

LWF LOBE_WIRE_FAULT   The interface has detected an open or short circuit in the cable between the interface and the wiring concentrator. The interface is closed and is at the state following initialization.

ARMV AUTO_REMOVAL_ERROR   The interface has failed the lobe wrap test, which resulted from the beacon auto-removal process, and has removed itself from the ring. The interface has closed and is at the state following initialization.

RMVD REMOVED_RECEIVED   The interface has received a remove ring station MAC frame request and has removed itself from the ring. The interface is closed and is at the state following initialization.

CO COUNTER_OVERFLOW   One of the following error counters has incremented from 254 to 255: Line, ARI/FCI, Frame Copy, Lost Frames, Burst, Lobe wire faults, Removes received. This display shows these error counters.

SSTA SINGLE_STATION   The interface has sensed that it is the only station on the ring.

RR RING_RECOVERY   The interface observes claim Token MAC frames on the ring. The interface may be transmitting the claim Token frames. This status remains until the interface transmits a ring purge frame.

Interface Restarts	Specifies the number of times the Token Ring chip timed out, or the Token Ring driver received a bad command from the handler. For information about why a restart occurred, see messages TKR.37, TKR.38, TKR.39, TKR.40, and TKR.41. in *elsmsg*.
# of times signal lost	Specifies the total number of times that the router was unable to transmit a packet due to loss of signal.
Hard errors	Displays the number of times the interface transmits or receives beacon frames from the network.
Auto-removal errors	Displays the number of times the interface, due to the beacon auto-removal process, fails the lobe wrap test and removes itself from the network.
Ring recovery actions	Displays the number of times the interface detects claim token medium access control (MAC) frames on the network.

Line errors	The Line Errors counter increments when a frame is repeated or copied and the Error Detected Indicator (EDI) is zero for the incoming frame:
	One of the following conditions must also exist:
	• A token with a code violation exists. • A frame has a code violation between the starting and ending delimiter. • A Frame Check Sequence (FCS) error occurs.
ARI/FCI errors	The ARI/FCI (Address Recognized Indicator/Frame Copied Indicator) Errors counter increments if the interface receives either of the following:
	An Active Monitor Present (AMP) MAC frame with the ARI/FCI bits equal to zero and a Standby Monitor Present (SMP) MAC frame with the ARI/FCI bits equal to zero.
	More than one SMP MAC frame with the ARI/FCI bits equal to zero, without an intervening AMP MAC frame.
	This error indicates that the upstream neighbor copied the frame but is unable to set the ARI/FCI bits.
Frame copy errors	Displays the number of times the interface in receive/repeat mode recognizes a frame addressed to its specific address but finds the address recognize indicator (ARI) bits not equal to zero. This error indicates a possible line hit or duplicate address.
Lost frames	Displays the number of times the interface is in transmit mode (stripping) and fails to receive the end of a transmitted frame.
# times beaconing	Displays the number of times the interface transmits a beacon frame to the network.
Lobe wire faults	Displays the number of times the network detects an open or short circuit in the cable between the interface and the wiring concentrator.
Removes received	Displays the number of times the interface receives a remove ring station MAC frame request and removes itself from the network.
Burst errors	Displays the number of times the interface detects the absence of transitions for five half-bit times between the start delimiter (SDEL) and the end delimiter (EDEL) or between the EDEL and the SDEL.
Inputs dropped	Displays the number of times an interface in repeat mode recognizes a frame addressed to it but has no buffer space available to copy the frame.
Token errors	The token errors counter increments when the active monitor detects a token protocol with any of the following errors:
	The MONITOR_COUNT bit of token with nonzero priority equals one.
	The MONITOR_COUNT bit of a frame equals one. No token or frame is received within a 10-ms window.
	The starting delimiter/token sequence has a code violation in an area where code violations must not exist.

Chapter 19. Configuring LLC Interfaces

This chapter describes how to set software configurable information for Link Level Control (LLC) interfaces in the router.

Link Level Control can be thought of as a "sub-protocol": It is not accessed directly from either the Talk 6 (configuration) or the Talk 5 (console) environment. Instead, it is accessed from the Token Ring, Point-to-Point (PPP), or Frame Relay protocols by entering an **LLC** command.

This chapter contains the following sections:

- "Accessing the Interface Configuration Process"
- "LLC Configuration Commands"

Accessing the Interface Configuration Process

Access the configuration commands for the protocol you wish to configure over LLC:

- Token Ring, as described in Chapter 17, "Configuring IEEE 802.5 Token-Ring Network Interfaces" on page 17-1
- Point-to-Point, as described in Chapter 28, "Using and Configuring Point-to-Point Protocol Interfaces" on page 28-1
- Frame Relay, as described in Chapter 26, "Using and Configuring Frame Relay Interfaces" on page 26-1

Each of these commands has an LLC command. Enter **LLC** to access the LLC configuration commands and perform LCC configuration. When you are finished, enter **Exit** to return to the prompt level for the protocol you are configuring.

LLC Configuration Commands

LLC configuration is required when you need to pass packets over an SNA network. To enter these commands, you must first enter the LLC configuration environment (see "Accessing the Interface Configuration Process" on page 17-1).

This section summarizes and then explains all of the LLC configuration commands. These commands (Table 19-1) enable you to configure LLC when you need to pass packets over a SNA network.

Table 19-1. LLC Configuration Command Summary	
Command	**Function**
? (Help)	Displays all the LLC commands or lists subcommand options for specific commands.
List	Displays the selected LLC configuration.
Set	Sets the timers associated with LLC, and the size of the transmit and receive windows.
Exit	Exits the LLC configuration process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
list
set
exit
```

List

Use the **list** command to display the current configuration for the LLC.

Syntax: l̲ist

Example: **list**

```
Reply Timer (T1):            1 seconds
Receive ACK Timer (T2):      100 milliseconds
Inactivity Timer (Ti):       30 seconds
Max Retry value (N2):        8
Rcvd I-frames before ACK (N3): 1
Transmit Window (Tw):        2
Receive Window (Rw):         2
Acks needed to increment Ww (Nw): 1
```

Reply Timer (T1)	This timer expires when the LLC fails to receive a required acknowledgment or response from the other LLC station.
Receive ACK Timer (T2)	This timer is used to delay sending of an acknowledgment for a received I-format frame.
Inactivity Timer (Ti)	This timer expires when the LLC does not receive a frame for a specified time period. When this timer expires the LLC transmits an RR until the other LLC responds or the N2 retry count is exceeded. Default is 30 seconds.
Max Retry value (N2)	The maximum number of retries by the LLC protocol. Default is 8.
Rcvd I-frames before ACK (N3)	This value is used with the T2 timer to reduce acknowledgment traffic for received I-frames. This counter sets a specified value and decrements each time an I-frame is received. When this counter reaches 0 or the T2 timer expires, an acknowledgment is sent. Default is 1.
Receive Window (Rw)	Indicates the maximum number of unacknowledged sequentially numbered I-frames that an LLC can receive from a remote host.
Transmit Window (Tw)	Indicates the maximum number of I-frames that can be sent before receiving an RR.
Acks needed to increment Ww (Nw)	This field is set to a default value of 1.

Set

Use the **set** command to configure the LLC.

Attention: Changing LLC parameters from the defaults can affect how the LLC protocol works.

Syntax: <u>se</u>t   <u>n2</u>-max-retry *count*
                  <u>n3</u>-frames-rcvd-before-ack *count*
                  <u>nw</u>-acks-to-inc-window *count*
                  <u>rw</u>-receive-window *count*
                  <u>t1</u>-reply-timer *seconds*
                  <u>t2</u>-receive-ack-timer *seconds*
                  <u>ti</u>-inactivity-timer *seconds*
                  <u>tw</u>-transmit-window *count*

n2-max-retry
> The maximum number of retries by LLC protocol. For example, N2 is the maximum number of times the LLC transmits an RR without receiving an acknowledgment when the inactivity timer expires. Default is 8. Minimum is 1. Maximum is 127.
>
> **Example: `set n2-max-retry`**
>
> `Max Retry value (N2) [8]?`

n3-frames_rcvd-before-ack
> This value is used with the T2 timer to reduce acknowledgment traffic for received I-frames. Set this counter to a specified value. Each time an I-frame is received, this value decrements. When this counter reaches 0 or the T2 timer expires, an acknowledgment is sent. Default is 1. Minimum is 1. Maximum is 255.
>
> **Example: `set n3-frames_rcvd-before-ack`**
>
> `Number I-frames received before sending ACK(N3) [1]?`

rw-receive-window
> Indicates the maximum number of unacknowledged sequentially numbered I-frames that an LLC can receive from a remote LLC peer. This value must be equal to or less than 127.
>
> **Example: `set rw-receive-window`**
>
> `Receive Window (Rw), 127 Max. [2]?`

nw-acks-to-inc-ww
> This field is set to a default value of 1.

t1-reply-timer
> This timer expires when the LLC fails to receive a required acknowledgment or response from the other LLC station. When this timer expires, an RR is sent with the poll bit set and T1 is started again. If the LLC receives no response after the configured maximum number of retries (N2), the link underneath is declared inoperative. Default is 1. Minimum is 1. Maximum is 256.
>
> **Example: `set t1-reply-timer`**
>
> `Reply Timer (T1) in sec. [1]?`

t2-receive-ack-timer
> This timer is used to delay sending of an acknowledgment for a received I-format frame. This timer is started when an I-frame is received. The timer is reset when an acknowledgment is sent. If this timer expires, LLC2 sends an

acknowledgment as soon as possible. Set this value so that it is less than that of T1. This insures that the remote LLC2 peer receives the delayed acknowledgment before the T1 timer expires. Default is 1 (100 ms). Minimum is 1. Maximum is 2560.

Example: `set t2-receive-ack-timer`

`Receive Ack timer (T2) in 100 millisec. [1]?`

Note: If this timer is set to 1 (the default) it will not run (for example, **n3-frames_rcvd-before-ack=1**).

ti-inactivity-timer
This timer expires when the LLC does not receive a frame for a specified time period. When this timer expires the LLC transmits an RR until the other LLC responds or the N2 retry count is exceeded. Default is 30 seconds. Minimum is 1 second. Maximum is 256 seconds.

Example: `set ti-inactivity-timer`

`Inactivity Timer (Ti) in sec. [30]?`

tw-transmit-window
Sets the maximum number of I-frames that can be sent before receiving an RR. Assuming that the other end of the LLC session can actually receive this many consecutive I-frames, and the router has enough heap memory to keep copies of these frames until an acknowledgment is received, increasing this value may increase the throughput. Default is 2. Minimum is 1. Maximum is 127.

Example: `set tw-transmit-window`

`Transmit Window (Tw), 127 Max. [2]?`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: <u>e</u>xit

Example: `exit`

Chapter 20. Monitoring LLC Interfaces

This chapter describes how to monitor specific LLC interfaces in the router by using either the interface console commands or the GWCON interface command.

Link Level Control can be thought of as a "sub-protocol": It is not accessed directly from either the Talk 6 (configuration) or the Talk 5 (console) environment. Instead, it is accessed from the Token Ring, Point-to-Point (PPP), or Frame Relay protocols by entering an **LLC** command.

This chapter includes the following sections:

* "Accessing the Interface Console Process"
* "LLC Monitoring Commands"

Accessing the Interface Console Process

Access the console commands for the protocol you wish to monitor over LLC:

* Token Ring, as described in Chapter 18, "Monitoring IEEE 802.5 Token-Ring Network Interfaces" on page 18-1
* Point-to-Point, as described in Chapter 29, "Monitoring Point-to-Point Protocol Interfaces" on page 29-1
* Frame Relay, as described in Chapter 27, "Monitoring Frame Relay Interfaces" on page 27-1

Each of these prompt-levels has an LLC command. Enter **LLC** to access the LLC console commands to monitor LCC. When you are finished, enter **Exit** to return to the prompt level for the protocol you are monitoring.

LLC Monitoring Commands

This section summarizes and then explains all of the LLC monitoring commands. These commands let you monitor the LLC while passing packets over an SNA network.

Table 20-1. LLC Monitoring Command Summary

Command	Function
? (Help)	Displays all the LLC commands or lists subcommand options for specific commands.
Clear-counters	Clears all statistical counters.
List	Displays interface, SAP, and session information.
Set	Allows the user to dynamically configure LLC parameters that are valid for the life of the session.
Exit	Exits the LLC monitoring process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
CLEAR-COUNTERS
LIST
SET
EXIT


set ?
```

Clear-Counters

Use the **clear-counters** command to clear all the LLC statistical counters.

Syntax: c̲lear-counters

Example: **clear-counters**

List

Use the **list** command to display interface, service access point (SAP), and session information.

Syntax: l̲ist i̲nterface
s̲ap . . .
se̲ssion

interface
   Displays all SAPs opened on this interface.

   Example: list interface

```
SAP      Number of Sessions
F4       1
```

sap sap_number
   Displays information for the specified SAP on the interface.

   Example: list sap

SAP value in hex (0FE) [1]? **F4**

```
Interface                     0, TKR/0
Reply Timer(T1)               1 sec
Receive ACK Timer (T2)        100 millisec
Inactivity Timer (Ti)         30 sec
MAX Retry Value (N2)          8
MAX I-field Size (N1)         2052
Rcvd I-frames before ACK (N3) 1
Transmit Window Size (Tw)     2
Acks Needed to Inc Ww (Nw)    1

Frame                         Xmt    Rcvd
UI-frames                     4      5
TEST-frames                   0      1
XID-frames                    0      0
I-frames                      291    26
RR-frames                     81     291
RNR-frames                    0      0
REJ-frames                    0      0
SABME-frames                  1      0
```

```
UA-frames                       0    1
DISC-frames                     0    0
DM-frames                       0    0
FRMR-frames                     0    0
I-frames discarded by LLC            0
I-frames Refused by LLC user         0

Cumulative number of sessions        1
Number of active sessions            1

Session ID                                      Remote
(int-sap-id)   Local MAC        Remote MAC       SAP    State
00F40000     00:00:C9:08:41:DB   10:00:5A:F1:02:37   F4   OPENED
```

SAP value in hex (0FE)	The SAP value of the session.
Interface	The interface number and type over which the session is running.
Reply Timer (T1)	Indicates the time it takes for this timer to expire when the LLC fails to receive an acknowledgment or response from the other LLC station.
Receive ACK Timer (T2)	Indicates the time delay the LLC uses before sending an acknowledgment for a received I-frame.
Inactivity Timer (Ti)	Indicates the time the LLC waits during inactivity before issuing an RR.
MAX Retry Value (N2)	The maximum number of retries by the LLC protocol.
MAX I-field Size (N1)	Maximum amount of data (in bytes) allowed in the I-field of an LLC2 frame.
Rcvd I-frame before ACK (N3)	Indicates the value that is used with T2 timer to reduce acknowledgment traffic for received I-frames.
Transmit Window Size (Tw)	Indicates the maximum number I-frames that can be sent before receiving an RR.
Acks Needed to Inc Ww (Nw)	This field is set to a default value of 1.
Frames Xmt and Rcvd	Counter that displays the total number of frame types transmitted (Xmt) and (Rcvd).
I-frames discarded by LLC	Counter that displays the total number of I-frames discarded by the LLC, usually because the sequence number is out of sequence.
I-frames refused by LLC user	Counter that displays the number of I-frames discarded by the software above the LLC. For example, DLSw (Data Link Switching).
Cumulative number of sessions	The total number of sessions that were opened over this SAP.
Number of active sessions	The total number of currently active sessions that are running over the interface.
Session ID (int-sap-id)	The session ID for the console interface.
Local MAC	The router's LLC MAC address.
Remote MAC	The remote LLC's MAC address.
Remote SAP	The remote SAP of the LLC connection.

Remote State	The finite state(s) that results from interaction between the LLC peers. There are 21 states that are described below.
Link_Closed	The remote LLC peer is not known to the local LLC peer and is considered as not existing.
Disconnected	The local LLC peer is known to the other peer. This LLC peer can send and receive XID, TEST, SABME, and DISC commands; and XID TEST, UA, and DM responses.
Link_Opening	The state of the local LLC peer after sending a SABME or UA in response to a received SABME.
Disconnecting	The state of the local LLC after sending a DISC command to the remote LLC peer.
FRMR_Sent	The local LLC peer has entered the frame reject exception state and has sent a FRMR response across the link.
Link_Opened	The local LLC peer is in the data transfer phase.
Local_Busy	The local LLC peer is unable to receive additional I-frames.
Rejection	A local LLC peer that has received one or more out-of-sequence I-frames.
Checkpointing	The local LLC peer has sent a poll to the remote LLC peer and is waiting for an appropriate response.
CKPT_LB	A combination of checkpointing and local busy states.
CKPT_REJ	A combination of the checkpointing and rejection states.
Resetting	The local LLC peer has received a SABME and is reestablishing the link.
Remote_Busy	The state that occurs when an RNR is received from the remote LLC peer.
LB_RB	A combination of local_busy and remote_busy states.
REJ_LB	A combination of rejection and local_busy states.
REJ_RB	A combination of rejection and remote_busy states.
CKPT_REJ_LB	A combination of checkpointing, rejection, and local_busy states.
CKPT_CLR	A combination state resulting from the termination of a local_busy condition while the LLC peer is CKPT_LB.
CKPT_REJ_CLR	A combination state resulting from the transfer of an unconfirmed local busy clear while the link station is in the CKPT_REJ_LB state.
REJ_LB_RB	A combination of the rejection, local_busy, and remote_busy states.
FRMR_Received	The local LLC peer has received an FRMR response from the remote LLC peer.

Session

Displays information on the specified LLC session that is open on the interface.

Example: `list session`

```
Session Id: [0]? 00-F4-0000

Interface0,                   TKR/0
Remote MAC addr               10:00:5A:F1:02:37
Source MAC addr               00:00:C9:08:35:47
Remote SAP                    F4
Local SAP                     F4
RIF                           (089E 0101 0022 0010)
Access Priority               0
State                         LINK_OPENED
Replay Timer                  1 sec
Receive ACK Timer (T2)        100 millisec
Inactivity Timer (Ti)         30 sec
MAX I-field Size (N1)         2052
MAX Retry Value (N2)          8
Rcvd I-frames before ACK (N3) 1
Transmit Window Size (Tw)     2
Working Transmit Size (Ww)    2
Acks Needed to Inc Ww (Nw)    1
Current Send Seq (Vs)         9
Current Rcv Seq (Vr)          7
Last ACK'd sent frame (Va)    9
No. of frames in ACK pend q   0
No. of frames in Tx pend q    0
Local Busy                    NO
Remote Busy                   NO
Poll Retry count              8
Appl output flow stopped      NO
Send process running          YES

Frame                         Xmt   Rcvd
I-frames                      1456  2678
RR-frames                     502   403
RNR-frames                    0     0
REJ-frames                    0     0
I-frames discarded by LLC           0
I-frames Refused by LLC user        0
```

Session Id	Indicates the session ID number.
Interface	Indicates the number of the interface over which this session is running.
Remote MAC addr	Indicates the MAC address of the remote LLC peer.
Source MAC addr	Indicates the MAC address of the local LLC.
Remote SAP	The remote side SAP of the LLC connection.
Local SAP	The local side SAP of the LLC connection.
RIF	The actual RIF of the frame.
Access Priority	Priority of the packet. 07 for upper layer control.
State	The finite state(s) that results from interaction between the LLC peers. Refer to the **list sap** command on page 20-2 for more information.
Receive ACK timer (T2)	Indicates the time delay the LLC uses before sending an acknowledgment for a received I-frame.
Inactivity timer (Ti)	Indicates the time the LLC waits during inactivity before issuing an RR.
MAX I-field size (N1)	Maximum size of the data field (in bytes) of a frame. Default is the size of the interface.

MAX Retry Value (N2)	The maximum number of times the LLC transmits an RR without receiving an acknowledgment
Rcvd I-frames before ACK (N3)	Indicates the value that is used with T2 timer to reduce acknowledgment traffic for received I-frames.
Transmit window size (Tw)	Indicates the maximum number of I-frames that can be sent before receiving an RR.
Working transmit size (Ww)	The maximum number of I-frames that are sent before receiving an RR.
Acks Needed to Inc Ww (Nw)	This field is set to a default value of 1.
Current send seq (Vs)	Send state variable (Ns value for the next I-frame to be transferred).
Current Rcv seq (Vr)	Receive state variable (next in-sequence Ns to be accepted).
Last ACK'd sent frame (Va)	Acknowledged state variable (last valid Nr received).
No. of frames in ACK pend q	Number of transmitted I-frames waiting for acknowledgment.
No. of frames in transmit pend q	Number of frames waiting to be transmitted.
Local Busy	The local side of the LLC connection is sending RNRs.
Remote Busy	The remote side of the LLC is receiving RNRs.
Poll Retry count	Indicates the current value of the retry of the counter (counts down) in the LLC protocol.
Appl output flow stopped	The LLC has told the application to stop giving it outgoing data frames.
Send process running	This process runs concurrently with all other frame actions and takes I-frames in the transmit queue and sends them.
Frames Xmt and Rcvd	Displays the total number of frame types transmitted (Xmt) and (Rcvd).
I-frames discarded by LLC	Counter that displays the total number of I-frames discarded by the LLC, usually because the sequence number is out of sequence.
I-frames refused by LLC user	Counter that displays the number of I-frames discarded by the software above the LLC. For example, DLSw (Data Link Switching).

Set

Use the **set** command to dynamically configure the LLC parameters on a current LLC session. Any changes that you make to the parameters are effective for the life of session. These parameters are the same as those listed in Chapter 17, "Configuring IEEE 802.5 Token-Ring Network Interfaces" on page 17-1.

Attention: Changing LLC parameters from the default can affect how the LLC protocol works.

Syntax: <u>se</u>t   <u>n2</u>-max_retry *count*
                <u>n3</u>-frames-rcvd-before-ack *count*
                <u>nw</u>-acks-to-inc-ww *count*
                <u>t1</u>-reply-timer *seconds*

> t2-receive-ack-timer *seconds*
> ti-inactivity-timer *seconds*
> tw-transmit-window *seconds*

n2-max_retry

The maximum number of retries by LLC protocol. For example, N2 is the maximum number of times the LLC transmits an RR without receiving an acknowledgment when the inactivity timer expires. Default is 8. Minimum is 1. Maximum is 127.

Example: `set n2-max_retry`

n3-frames-rcvd-before-ack

This value is used with the T2 timer to reduce acknowledgment traffic for received I-frames. Set this counter to a specified value. Each time an I-frame is received, this value is decremented. When this counter reaches 0 or the T2 timer expires, an acknowledgment is sent. Default is 1. Minimum is 1. Maximum is 255.

Example: `set n3-frames-rcvd-before-ack`

nw-acks-to-inc-ww

This field is set to a default value of 1.

t1-reply-timer

This timer expires when the LLC fails to receive a required acknowledgment or response from the other LLC station. When this timer expires, an RR is sent with the poll bit set and T1 is started again. If the LLC receives no response after the configured maximum number of retries (N2), the link underneath is declared inoperative. Default is 1. Minimum is 1. Maximum is 256.

Example: `set t1-reply-timer`

t2-receive-ack-timer

This timer is used to delay sending of an acknowledgment for a received I-format frame. This timer is started when an I-frame is received and reset when an acknowledgment is sent. If this timer expires, LLC2 sends an acknowledgment as soon as possible. Set this value so that it is less than that of T1. This insures that the remote LLC2 peer receives the delayed acknowledgment before the T1 timer expires. Default is 1 (100 ms). Minimum is 1. Maximum is 2560.

Example: `set t2-receive-ack-timer`

Note: If this timer is set to 1 (the default) it will not run (for example, **n3-frames-rcvd-before-ack=1**).

ti-inactivity-timer

This timer expires when the LLC does not receive a frame for a specified time period. When this timer expires the LLC transmits an RR until the other LLC responds or the N2 timer expires. Default is 30 seconds. Minimum is 1 second. Maximum is 256 seconds.

Example: `set ti-inactivity-timer`

tw-transmit-window

Sets the maximum number of I-frames that can be sent before receiving an RR. Assuming that the other end of the LLC session can actually receive this many consecutive I-frames, and the router has enough heap memory to keep copies of these frames until an acknowledgment is received, increasing this

value may increase the throughput. Default is 2. Minimum is 1. Maximum is 127.

Example: `set tw-transmit-window`

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: `exit`

Chapter 21. Configuring the Ethernet Network Interface

This chapter describes how to configure the Ethernet interface. It includes the following sections:

- "Accessing the Interface Configuration Process"
- "Ethernet Configuration Commands" on page 21-2

Accessing the Interface Configuration Process

Use the following procedure to access the configuration process. This process gives you access to an Ethernet interface's *configuration* process.

1. At the OPCON prompt (*), enter the **status** command to find the PID for CONFIG. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter **talk 6**. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

   ```
   * talk 6
   Config>
   ```

   After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

3. At the CONFIG prompt, enter the **list devices** command to display the network interface numbers for which the router is currently configured.

4. Record the interface numbers.

5. Enter the **network** command and the number of the Ethernet interface you want to configure. For example:

   ```
   Config> network 0
   ETH Config>
   ```

   The Ethernet configuration prompt (`ETH Config>`), is displayed.

Ethernet Configuration Commands

This section summarizes and then explains the Ethernet configuration commands. Enter the commands at the `ETH config>` prompt.

Command	Function
Table 21-1. Ethernet Configuration Command Summary	
Command	**Function**
? (Help)	Displays all the Ethernet commands or lists subcommand options for specific commands.
Connector-Type	Sets the connector type.
Frame	Sets the NetWare IPX encapsulation type.
IP-Encapsulation	Sets the IP encapsulation as Ethernet (type X'0800') or IEEE (802.3 with SNAP).
List	Displays the current connector-type, NetWare IPX encapsulation, and IP encapsulation.
Exit	Exits the Ethernet config process.

? (Help)

Use the **?** (**help**) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: `ETH config>` **?**

Connector-Type

Use the **connector-type** command to set the connector type. 2216s support AUI (10BASE5) and RJ-45 (10BASE-T) connectors, and auto-config options.

Syntax: c̲onnector-type *name*

Example: `connector-type aui`

Frame

Use the **frame** command to select the Ethernet encapsulation format used by this interface. This is required if you are using NetWare-VMS on the Ethernet, and is often used when there are ISO nodes on the same Ethernet. The following options are available:

- ethernet_II (default of NetWare 4.0 and greater) - Ethernet_II uses Ethernet version 2.0 protocol 81-37.

- ethernet_8022 - Ethernet_8022 uses Ethernet 802.3 with 802.2 SA E0.

- ethernet_8023 (default of pre-NetWare 4.0 and lower) - Ethernet_8023 uses Ethernet 802.3 without any 802.2 header.

- ethernet_SNAP - Ethernet_SNAP uses 802.3, 802.2 with SNAP PID 00-00-00-81-37.

Note: The ethernet_SNAP encapsulation is not architecturally valid and is not fast-pathed. No cache entries will appear for network entries using this encapsulation.

Syntax: frame *type*

Example: `frame ethernet_II`

IP-Encapsulation

Use the **IP-encapsulation** command to select Ethernet (Ethernet type X'0800') or IEEE 802.3 (Ethernet 802.3 with SNAP). Enter **e** or **i**.

Syntax: IP-encapsulation *type*

Example: `IP-encapsulation e`

List

Use the **list** command to display the current configuration for the Ethernet interface including the connector-type, IPX encapsulation type, and IP encapsulation type.

Syntax: l̲ist   a̲ll

Example: `list all`

```
          Connector type:          AUI (10BASE5)
          NetWare IPX encapsulation: ETHERNET_SNAP
          IP Encapsulation:        ETHER
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: `exit`

Chapter 22. Monitoring the Ethernet Network Interface

This chapter describes how to monitor the Ethernet interfaces. It includes the following sections:

- "Displaying Ethernet Statistics through the Interface Command"
- "Accessing the Interface Console Process" on page 22-4
- "Ethernet Interface Console Commands" on page 22-4

Displaying Ethernet Statistics through the Interface Command

You can also use the **interface** command from the GWCON environment to display the following statistics.

```
+interface 4
                                     Self-Test  Self-Test  Maintenance
Nt Nt'  Interface  Slot-Port         Passed     Failed     Failed
4  4    Eth/0      Slot: 4  Port: 1     3          1          0

 Ethernet/IEEE 802.3 MAC/data-link on Ethernet interface

 Physical address        AA0004000318
 PROM address            10005AFF0016
 Microcode Level         Uu17c

 Input statistics:
  failed, packet too long         0   failed, CRC error            1
  failed, alignment error         0   failed, FIFO over-run        0
  buffer full warnings            0   packets missed               1
  internal mac rx errors          0
 Output statistics:
  initially deferred             12   single collision             1
  multiple collisions             1   total collisions             4
  failed, excess collisions       0   failed, FIFO under-run       0
  failed, carrier check           0   CD heartbeat error           0
  out of window colls             1
```

The next section describes the above output.

Nt
    Global network number.

Nt'
    This field is for the serial interface card. Disregard the output.

Interface
    Interface name and its instance number.

Port
    Port number

Slot
    Slot number

Self-Test: Passed
    Number of self-tests that succeeded.

Self-Test: Failed
    Number of self-tests that failed.

Maintenance: Failed
> Number of maintenance failures.

Physical address
> The Ethernet address of the device currently in use. This may be the PROM address or an address overwritten by some other protocol.

PROM address
> The permanent unique Ethernet address in the PROM for this Ethernet interface.

Interface restarts
> The number of times the Ethernet chip timed out, or the Ethernet driver received a bad command from the handler. For information about why a restart occurred, refer to messages Eth.043 and Eth.044 in the *IBM Nways Event Logging System Messages Guide*.

Interface type
> This specifies the connector type as AUI or RJ45.

Input statistics:

failed, packet too long   or   failed, frame too long
> The Failed, Packet Too Long counter increments when the interface receives a packet that is larger than the maximum size of 1518 bytes for an Ethernet frame. This data is exported via SNMP as the dot3StatsFrameTooLongs counter.

failed, CRC error or failed, FCS (Frame Check Sequence) error
> The Failed, CRC (Cyclic Redundancy Check) Error counter increments when the interface receives a packet with a CRC error. This data is exported via SNMP as the dd3StatsFCSErrors counter.

failed, framing error or failed, alignment error
> The Failed, Framing Error counter increments when the interface receives a packet whose length in bits is not a multiple of eight.

failed, FIFO over-run or failed, FIFO overrun
> The Failed, FIFO (First In, First Out) Overrun counter increments when the Ethernet chipset is unable to store bytes in the local packet buffer as fast as they come off the wire.

collision in packet
> The counter increments when a packet collides as the interface attempts to receive a packet, but the local packet buffer is full. This error indicates that the network has more traffic than the interface can handle.

short frame
> The counter increments when the interface receives a packet with a short frame.

buffer full warnings
> The Buffer Full Warnings counter increments each time the local packet buffer is full.

packets missed
> The Packets Missed counter increments when the interface attempts to receive a packet, but the local packet buffer is full. This error indicates that the network has more traffic than the interface can handle.

internal mac rcv errors
> Receive errors that are not late, excessive, or carrier check collisions. This data is exported via SNMP as the dot3StatsInternalMacReceiveErrors counter. This statistic is the sum of the FIFO Overruns.

Output statistics:

initially deferred or deferred transmission
> The Initially Deferred counter increments when the carrier sense mechanism detects line activity causing the interface to defer transmission. This data is exported via SNMP as the dot3StatsDeferredTransmissions counter.

single collision
> The Single Collision counter increments when a packet has a collision on the first transmission attempt, and then successfully sends the packet on the second transmission attempt. This data is exported via SNMP as the dot3StatsSingleCollisionFrames counter.

multiple collisions
> The Multiple Collisions counter increments when a packet has multiple collisions before being successfully transmitted. This data is exported via SNMP as the dot3MultipleCollisionFrames counter.

total collisions
> The Total Collisions counter increments by the number of collisions a packet incurs.

failed, excess collisions
> The Failed, Excess Collisions counter increments when a packet transmission fails due to 16 successive collisions. This error indicates a high volume of network traffic or hardware problems with the network. This data is exported via SNMP as the dot3StatsExcessiveCollisions counter.

failed, FIFO underrun
> The Failed, FIFO Underrun counter increments when packet transmission fails due to the inability of the interface to retrieve packets from the local packet buffer fast enough to transmit them onto the network.

failed, carrier check or failed, carrier sense error
> The Failed, Carrier Check counter increments when a packet collides because carrier sense is disabled. This error indicates a problem between the interface and its Ethernet transceiver. This data is exported via SNMP as the dot3StatsCarrierSenseErrors counter.

CD heartbeat error or SQE test error
> The CD (Collision Detection) Heartbeat Error or SQE (Signal Quality Error) counter increments when the interface sends a packet but detects that the transceiver has no heartbeat. The packet is treated as successfully transmitted because some transceivers do not generate heartbeats. This data is exported via SNMP as the dot3StatsSQETestErrors counter.

out of window collisions or late collisions
> The Out of Window Collisions counter increments when a packet collides after transmitting at least 512 bits. This error indicates that an interface on the network failed to defer, or that the network has too many stations.

internal mac tx errors or internal MAC trans errors
> Transmit errors that are not late, excessive, or carrier check collisions. This data is exported via SNMP as the dot3StatsInternalMacTransmitErrors counter. This statistic is the sum of the FIFO Underruns.

RISC Microcode Version:
This gives the version of the microcode running in the RISC controller of the communications processor module.

Accessing the Interface Console Process

To monitor information related to the Ethernet Network Interface, access the interface console process by doing the following:

1. Enter the **status** command to find the PID for GWCON. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter **talk 5**. For example:

   ```
   * talk 5
   ```

   The GWCON prompt (+) is displayed on the console. If the prompt does not appear when you first enter GWCON, press **Return** again.

3. At the GWCON prompt, enter the **configuration** command to see the protocols and networks for which the router is configured. For example:

   ```
   + configuration
   ```

   See page 6-4 for sample output of the **configuration** command.

4. Enter the **network** command and the number of the Ethernet interface. In this example:

   ```
   + network 0
   ETH>
   ```

   The Ethernet console prompt is displayed. You can now view information about the Ethernet interface by entering console commands.

Ethernet Interface Console Commands

This section summarizes and explains the Ethernet console commands. Enter commands at the ETH> prompt. Table 22-1 lists the console commands.

Table 22-1. Ethernet Console Command Summary	
Command	**Function**
? (Help)	Displays all the Ethernet commands or lists subcommand options for specific commands.
Collisions	Displays collision statistics for the specified Ethernet interface.
Exit	Exits the Ethernet monitoring process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: Eth> **?**

```
collisions
exit
```

Collisions

This command shows the counts of transmissions for packets that incurred collisions before successful transmission. Counters are given for packets sent after the collision XXXXx packets sent after 15 collisions. Increasing numbers of packets transmitting with collisions and higher numbers of collision per packet are signs of transmitting onto a busy Ethernet.

These counters are cleared by the OPCON **CLEAR** command. This data is exported via SNMP as the dot3CollTable counter.

Syntax: <u>coll</u>isions

Example: Eth> **coll**

```
Transmitted with  1 collisions:0
Transmitted with  2 collisions:0
Transmitted with  3 collisions:0
Transmitted with  4 collisions:0
Transmitted with  5 collisions:0
Transmitted with  6 collisions:0
Transmitted with  7 collisions:0
Transmitted with  8 collisions:0
Transmitted with  9 collisions:0
Transmitted with 10 collisions:0
Transmitted with 11 collisions:0
Transmitted with 12 collisions:0
Transmitted with 13 collisions:0
Transmitted with 14 collisions:0
Transmitted with 15 collisions:0
```

Exit

Use the **exit** command to return to the previous prompt level (GWCON).

Syntax: <u>e</u>xit

Example: Eth> **exit**

Chapter 23. Configuring Serial Line Interfaces

This chapter describes the interface configuration process for a serial interface and includes the following sections:

- "Accessing the Interface Configuration Process"
- "Network Interfaces and the GWCON Interface Command"

IMPORTANT: To configure Frame Relay, PPP, X.25, V.25bis, SDLC Relay, and SDLC protocols on the serial interface, use the commands in this chapter and then refer to the commands in the chapters that describe the specific protocol.

See "Configuring the Network Interface" on page 1-18 for a table of protocols and the interfaces that support those protocols.

Accessing the Interface Configuration Process

See "Accessing Network Interface Configuration and Console Processes" on page 1-15 for a description of how to add a serial interface. Once you have done that, the following paragraphs describe how to set the data-link of the interface correctly and how to access that data-link's configuration commands.

To access the interface configuration process for a serial interface, first access the `Config>` prompt and issue the command **set data-link**. Next, at the `Config>` prompt, enter the interface type and number to access the configuration environment for the interface.

For example, to configure a serial interface for X.25, you must access the `X.25 config>` environment by issuing the following commands:

```
Config> set data-link X25 2
Config> network 2
```

From the `X.25 config>` environment, you can complete your configuration of X.25 on the serial interface. See Chapter 24, "Configuring the X.25 Network Interface" on page 24-1.

When you are done configuring the serial interface, enter the **restart** command after the OPCON prompt (*) and respond **yes** to the prompt to enable the new configuration.

Network Interfaces and the GWCON Interface Command

While serial line interfaces do not have their own console process for monitoring purposes, routers can display complete statistics for all installed network interfaces when you use the **interface** command from the GWCON environment. For more information on the **interface** command and displaying statistics, see Chapter 6, "The GWCON (Monitoring) Process and Commands."

Chapter 24. Configuring the X.25 Network Interface

The X.25 network interface connects a router to an X.25 virtual circuit switched network. The X.25 network interface software and hardware allows the router to communicate over a public X.25 network. The X.25 network interface complies with CCITT 1980, CCITT 1984, CCITT 1988 and ISO 8208 1990 specifications for X.25 interfaces offering multiplexed channels and reliable end-to-end data transfer across a wide area network.

This chapter includes the following sections:

- "Basic Configuration Procedures"
- "X.25 Configuration Commands" on page 24-4

For information on configuring X.25 Transport Protocol (XTP) for transporting X.25 traffic over TCP/IP, refer to *Protocol Configuration and Monitoring Reference Volume 2 for Nways Multiprotocol Access Services Version 1 Release 1*.

Basic Configuration Procedures

This section outlines the minimal configuration steps required to get the X.25 interface up and running. The X.25 parameters must be consistent with the X.25 network the interface on the router will connect to. For more information, refer to the configuration commands described in this chapter.

Note: You must restart the router for the configuration changes to take effect.

1. At the OPCON prompt (*), type **talk 6**.

   The Config> prompt appears.

2. Type **list devices** to display a list of the interfaces from which you can select. Use the appropriate interface number in the following step.

3. Type **set data-link x25**.

   The Interface Number [0]? prompt appears.

4. Type the appropriate interface number.

5. Connect to the network by typing **net** *network#* at the Config> prompt.

   The X.25 Config> prompt appears.

6. At this prompt, type **set address** *x.25-node-address*.

   The X.25 address is a unique X.121 address that is used during call establishment. You **must** specify an X.25 node address for each router interface. Failure to set the network address prevents the X.25 interface from joining the attached network.

7. Type **set equipment-type** and specify whether the frame and packet levels act as DCE or DTE. The default for this command is DTE.

8. Type **set svc** and define the lowest and highest SVCs that you are using. The default is for 1 SVC.

9. Type **add protocol** *protocol_name* to add the protocols that will be running over the X.25 interface. You will be prompted for window size, default packet size, maximum packet size, circuit idle time, and max VCs.

> **Note:** You need to add the protocols only once for all X.25 networks on the router.

10. Type **add address** *protocol_name* to add an address translation for each protocol's destination address reachable over this interface.

11. Type **exit** to return to the `Config>` prompt.

12. Press **Ctrl** **P** to return to the OPCON prompt (*).

13. Type **restart** and respond **yes** to the prompt.

Setting the National Personality

Each public data network, such as GTE's Telenet or DDN's Defense Data Network, has its own standard configuration. The term *National Personality* specifies a group of variables used to define a public data network's characteristics. The configuration information in the National Personality provides the router with control information for packets being transferred over the link. The National Personality option defines 28 default parameters for each public data network.

To view the configuration values that are in your X.25 National Personality, execute the X.25 configuration **list detailed** command. Configure each public data network connected to the router by executing the X.25 configuration **set national personality** command.

The National Personality is a generalized template for network configuration. If necessary, you can individually configure each frame and packet layer parameter.

Understanding the X.25 Defaults

The following tables list the defaults for the various parameters for the X.25 *set, national set* and *national enable* commands.

Table 24-1 (Page 1 of 2). Set Command

Parameter	Default
address ...	none
cable	none
calls-out ...	4
clocking ...	external
default-window-size ...	2
encoding	NRZ
equipment-type ...	DTE
htf addr ...	none
inter-frame-delay ...	0
mtu	1500
national-personality ...	GTE Telenet
pvc ...	low=0 high=0
speed ...	9600
svc	low inbound=0, high inbound=0 low 2-way=1, high 2-way=64 low outbound=0, high outbound=0

Table 24-1 (Page 2 of 2). Set Command

Parameter	Default
throughput-class ...	inbound=outbound=2400
vc-idle ...	30

Table 24-2. National Enable Parameters

Parameter	DDN Default	GTE Default
accept-reverse-charges	off	on
ccitt	on	on
flow-control-negotiation	on	on
frame-ext-seq-mode	off	off
osi-84	on	on
osi-88	off	off
packet-ext-seq-mode	off	off
request-reverse-charges	off	on
suppress-calling-addresses	off	off
throughput-class-negotiation	off	on

Table 24-3. National Set Parameters

Paramter	DDN Default	GTE Default
call-req	20 decaseconds	20 decaseconds
clear-req ...	retries=1	retries=1
	18 decaseconds	18 decaseconds
disconnect-procedure ...	passive	passive
dp-timer	500 milliseconds	500 milliseconds
frame-window-size	7	7
network-type ...	DDN	CCITT
n2-timeouts	20	20
packet-size ...	128, max=256	128, max=256
reset ...	retries=1	retries=1
	18 decaseconds	18 decaseconds
restart ...	retries=1	retries=1
	18 decaseconds	18 decaseconds
min-recall	10 seconds	10 seconds
min-connect	90 seconds	90 seconds
collision-timer	10 seconds	10 seconds
standard-version	1984	1984
t1-timer	4 seconds	4 seconds
t2-timer	0	0

X.25 Configuration Commands

This section summarizes and explains all the X.25 configuration commands.

The X.25 configuration commands allow you to specify network parameters for router interfaces that transmit X.25 packets. The information you specify with the configuration commands activates when you restart the router.

Enter the X.25 configuration commands at the X.25 config> prompt. Table 24-4 shows the commands.

Table 24-4. X.25 Configuration Commands Summary	
Command	**Function**
? (Help)	Lists the interface configuration commands or lists the options associated with specific commands.
Set	Sets the local and DDN X.25 node addresses, window size for frame and packet levels, identifies the National personality, the MTU, and the maximum number of calls. Defines the PVC and SVC channel ranges, the number of seconds that a switched circuit can be idle before it is cleared, and specifies whether one router needs to act as a DCE (when two routers are directly connected without an intervening X.25 network) or the more normal method of acting at a DTE connected to an X.25 network.
Enable/Disable	Enables/Disables incoming-calls-barred feature, outgoing-calls-barred feature, dynamic DDN address translations, and lower-dtr feature.
National Enable or National Disable	Enables/Disables the parameters defined by the National Personality configuration.
National Set	Sets parameters defined by the National Personality configuration.
National Restore	Restores the National Personality configuration to its default values.
Add/Change/Delete	Adds/Changes/Deletes an address translation, a protocol encapsulation, or a PVC definition.
List	Lists the defined address translations, National Personality parameters, protocol encapsulation, or PVC definitions.
Exit	Exits the X.25 configuration process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

```
                      Add
                      Change
                      Delete
                      Disable
                      Enable
                      List
                      National-Personality
                      Set
                      Exit
```

Set

Use the **set** command to configure local X.25 node addresses, maximum number of calls, frame and packet level window size, lowest to highest PVC and SVC channels, and the idle time for a switched circuit.

Syntax: <u>s</u>et   <u>a</u>ddress . . .
                       <u>cab</u>le
                       <u>cal</u>ls-out . . .
                       <u>c</u>locking . . .
                       <u>d</u>efault-window-size . . .
                       <u>e</u>ncoding
                       <u>e</u>quipment-type . . .
                       <u>h</u>tf addr . . .
                       <u>i</u>nter-frame-delay . . .
                       <u>m</u>tu
                       <u>n</u>ational-personality . . .
                       <u>p</u>vc . . .
                       <u>sp</u>eed . . .
                       <u>sv</u>c
                       <u>t</u>hroughput-class . . .
                       <u>v</u>c-idle . . .

address   *x.25-node-addr*

    Sets the local X.25 interface address (*x.25-node-addr*). Set the X.25 node address to 0, not to 00, to delete the local X.25 address.

    Example: `set address 8982800`

cable *type*

    Sets the cable type as follows:

- RS-232 DTE
- RS-232 DCE
- V35 DTE
- V35 DCE
- V36 DCE
- V36 DTE
- X21 DTE
- X21 DCE

    A DTE cable is used when you are attaching the router to some type of DCE device (for example, a modem or a DSU/CSU).

    A DCE cable is used when the router is acting as the DCE and providing the clocking for direct attachment.

Notes:

1. If you are configuring an interface on the 8-port EIA 232 adapter then the only cable types you can configure are RS-232 DTE and RS-232 DCE.

2. If you are configuring an interface on the 6-port V.35/V.36 adapter then the only cables types you can configure are V.35 DTE, V.35 DCE, V.36 DTE, V.36 DCE.

3. If you are configuring an interface on the 8-port X.21 adapter then the only cable types you can configure are X.21 DTE and X.21 DCE.

Example: `set cable RS-232 DTE`

calls-out   *value*
Sets the maximum number of locally initiated, simultaneously active SVCs.

Valid Values: 1 to 239

Default Value: 4

Example: `set calls-out 3`

clocking   *external* or *internal*
To connect to a modem or DSU, configure clocking as external. To connect directly to another DTE device, use a DCE cable, set the clocking to internal, and configure the line speed. For internal clocking, the line speed depends on the interface, as shown in Table 24-5 on page 24-8.

Default: external

Example: `set clocking internal`

default-window-size   *svalue*
Sets the window size for the packet level assigned by the router if there is no window-size facility in the Call-Request packet. The range is determined by the National Personality packet modulus (PACKET-EXT-SEQ-MODE).

Default: 2

Example: `set default-window-size 3`

encoding   *NRZ* OR *NRZI*
Sets the HDLC transmission encoding scheme for the interface. Encoding may be set for NRZ (non-return to zero) or NRZI (non-return to zero inverted). NRZ is the more widely used encoding scheme while NRZI is used in some IBM configurations.

Default: NRZ

Example: `set encoding nrz`

equipment-type   *DCE* OR *DTE*
Specifies whether the frame and packet levels act as DCE or DTE. This command has no relation to the cable type in use.

Default: DTE

Example: `set equipment-type DCE`

htf addr   *x.25-node-addr*
Sets the local X.121 address translation (*x.25-node-addr*). Set the X.25 node address to 0.0.0.0 to delete the local X.25 node address.

Example: `set htf-address 11.42.0.137`

inter-frame-delay *value*
> This parameter defines the minimum delay between transmitted frames. Setting this parameter is useful when interfacing directly to older equipment which may not be able to consistently handle consecutive frames separated by one flag (resulting in receive errors such as T1 timeouts).
>
> The 2216 requests from 0 to 15 extra flags between frames.
>
> **Default:** 0
>
> **Example:** `set inter-frame-delay 1`

mtu *value*
> Sets the Maximum Transmit Unit (MTU) in bytes. This is the maximum message size that will be delivered to the X.25 interface to package and transmit over the serial line. The range is 576–4096.
>
> **Default:** 1500
>
> If you are encountering packet reassembly timeouts when transferring data over the X.25 interface, you should determine what the minimum packet size is for all LAN or serial interfaces that lead to the end-point, then calculate a more suitable X.25 MTU. You should not directly consider the actual X.25 packet size in this calculation because X.25 tends to use a smaller packet size. X.25 usually sends up to 7 packets at one time before waiting for an acknowledgement.
>
>> For example, consider a network topology that includes:
>>
>> - A Token-Ring LAN having a packet size of 4000
>> - An X.25 serial line having a packet size of 128 with a window size of 7 and a bit rate of 9600 bps
>> - An Ethernet LAN with a packet size of 1500
>
>> In this case, you should probably set the X.25 MTU to 1500. That means that about 12 packets will be sent over the X.25 interface. (MTU / X.25 packet size = number of X.25 packets to be sent).
>
>> When using an MTU of 4096, 32 packets must be sent over the X.25 interface. (4000 /128 = 31.25). In this case, packet reassembly timeouts will probably occur if the X.25 modem speed is 9600 bps. Using an X.25 modem speed of 56 Kbps would probably solve this problem.
>
> **Example:** `set mtu 2048`

national-personality *GTE-Telenet* or *DDN*
> Sets the 28 default parameters for either GTE-Telenet or DDN National Personality.
>
> **Default:** GTE-Telenet
>
> **Example:** `set national-personality DDN`

pvc low/high *value*
> Defines the lowest to the highest Permanent Virtual Circuit channel number. Zero indicates no PVCs. By default there are no PVCs.
>
> **pvc low** 0
>
> **pvc high** 0
>
> The range is 1 to 4095. These values are setting boundaries of a given VC range. There is a maximum of 239 VCs in the 2216. There is a maximum of 200 PVCs.

Example: `set pvc low 40`

Note: Values must not overlap values set for SVCs.

speed *speed-setting*
For internal clocking, this command specifies the speed of the transmit and receive clock lines. Use Table 24-5 to determine the clock speeds you can set for the various adapters.

Table 24-5. Line Speeds When Internal Clocking is Used for 2216 Interfaces	
Adapter Type	**Speed Range**
8-port EIA 232	9600 to 64 000 bps
6-port V.35/V.36	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps
8-port X.21	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps

For external clocking, this command does not affect the hardware but it sets the speed some protocols, such as IPX, use to determine routing cost parameters. In these cases, set the speed to match the actual line speed. The maximum line speed that can be configured if external clocking is shown in Table 24-6.

Table 24-6. Line Speeds When External Clocking is Used for 2216 Interfaces	
Adapter Type	**Speed Range**
8-port EIA 232	2400 to 64 000 bps
6-port V.35/V.36	2400 to 2 048 000 bps
8-port X.21	2400 to 2 048 000 bps

Default: 9600

Note: The X.25 software is supported only at speeds up to 256000 bps.

Example: `set speed 19200`

svc low/high *inbound* OR *two-way* OR *outbound value*
Defines the lowest to the highest switched virtual circuit channel number. When low=high=0, no VCs in this category are defined.

Example: `set SVC low-two-way 1`

Inbound Specifies a range 1–255. By default, there are no inbound-only SVCs.

Valid values: 0-4095

Default values: 0

Two-way Specifies a range 1–255. By default, there are sixty-four 2-way SVCs.

Valid values: 0-4095

Default values:

svc low   1

svc high   64

Outbound Specifies a range 1–255. By default, there are no outbound-only SVCs.

> **Valid values:** 0-4095

> **Default:** 0

Note: Values in each range must not overlap other SVC ranges nor the PVC range.   Table 24-7 shows a possible VC configuration.

Table 24-7. Example VC Definitions. Only 239 VCs can be used from this definition

	Low	**High**
PVC	1	40
inbound	0	0
two-way	41	59
outbound	60	500

throughput-class inbound OR outbound *bit-rate*
Defines the throughput class requested when making a call request while throughput negotiation is enabled.

Default: 2400 bps

This setting is ignored when processing incoming `call request`s.

Example: `set throughput-class inbound`

vc-idle *value*
Defines the number of seconds that a switched circuit can be idle before it is cleared by the router. Zero indicates that the router never clears an idle circuit.

Valid values: 1 to 255

Default: 30 seconds

Example: `set vc-idle 40`

Enable

Use the **enable** command to enable DDN address translations, interface resets, or the incoming-calls-barred, outgoing-calls-barred, and lower-dtr features.

Syntax: <u>e</u>nable   <u>d</u>dn-address-translations
<u>inc</u>oming-calls-barred
<u>l</u>ower-dtr
<u>o</u>utgoing-calls-barred

ddn-address-translations
Specifies that the DDN host table format (IP) addresses are dynamically converted to X.121 call addresses. The default is Disabled.

Example: `enable ddn-address-translations`

incoming-calls-barred
Specifies that the router will not accept incoming calls. The default setting for this parameter is disabled or *off*, which allows incoming calls.

Example: `enable incoming-calls-barred`

lower-dtr
> This parameter determines the way the data terminal ready (DTR) signal is handled for leased serial-line interfaces that are disabled. If this parameter is set to "disabled" (the default), the DTR signal will be raised when the interface is disabled.
>
> If *lower-dtr* is set to "enabled," the DTR will be lowered when the interface is disabled. This behavior may be desirable in situations where the interface has been configured as an alternate link for WAN Reroute and the interface is connected to a dial-out modem which maintains its dial connection based on the state of the DTR signal.
>
> When lower-dtr is enabled and the interface is disabled, the DTR signal is low and the modem keeps the dial connection down. When the interface is enabled, due to a WAN Reroute backup scenario, DTR is raised and the modem dials a stored number to the backup site. When the primary interface is restored, the alternate interface is disabled, DTR is lowered, and the modem hangs up the dial connection.
>
> The following cable types are supported:
>
> > RS-232
> > V.35
> > V.36
>
> The default setting is disabled.
>
> **Example:** `enable lower-dtr`

outgoing-calls-barred
> Specifies that the router will not allow outgoing calls. The default setting for this parameter is disabled or *off*, which allows outgoing calls.
>
> **Example:** `enable outgoing-calls-barred`

Disable

Use the **disable** command to disable DDN address translations, interface resets as part of network certification, or the incoming-calls-barred or outgoing-calls-barred features.

Syntax: <u>dis</u>able   <u>dd</u>n-address-translations
                   <u>inc</u>oming-calls-barred
                   <u>l</u>ower-dtr
                   <u>o</u>utgoing-calls-barred

National Enable

Use the **national enable** command to enable a feature defined in the National Personality configuration.

Syntax: <u>n</u>ational <u>e</u>nable   <u>acc</u>ept-reverse-charges
                               <u>cc</u>itt
                               <u>fl</u>ow-control-negotiation
                               <u>fr</u>ame-ext-seq-mode
                               <u>osi-84</u>
                               <u>osi-88</u>
                               <u>packet-ext</u>-seq-mode
                               <u>req</u>uest-reverse-charges
                               <u>suppress-</u>calling-addresses

<u>t</u>hroughput-class-negotiation

accept-reverse-charges
> Accepts reverse charge calls during call establishment. This option is not available for DDN.
>
> **DDN Default**   off
>
> **GTE Default**   on
>
> **Example:**   `national enable accept-reverse-charges`

ccitt
> Specifies that the CCITT convention, rather than ISO convention, will be followed for timer retry expiration. CCITT acts as if the confirmation packet for the restart or clear requests had arrived. ISO leaves the request unconfirmed.
>
> **DDN Default**   on
>
> **GTE Default**   on
>
> **Example:**   `national enable ccitt`

flow-control-negotiation
> Enables the negotiation of packet and window size during call setup of SVCs.
>
> **DDN Default**   on
>
> **GTE Default**   on
>
> **Example:**   `national enable flow-control-negotiation`

frame-ext-seq-mode
> Sets the frame layer sequence numbering to modulo 128 (i.e., 0 through 127).
>
> **DDN Default**   off
>
> **GTE Default**   off
>
> **Example:**   `national enable frame-ext-seq-mode`

osi-84
> Allows CCITT OSI facilities defined by the 1984 standard.
>
> **DDN Default**   on
>
> **GTE Default**   on
>
> **Example:**   `national enable osi-84`

osi-88
> Allows CCITT OSI facilities defined by the 1988 standard.
>
> **DDN Default**   off
>
> **GTE Default**   off
>
> **Example:**   `national enable osi-88`

packet-ext-seq-mode
> Enables the packet layer to use extended sequence numbers (0 through 127).
>
> **DDN Default**   off
>
> **GTE Default**   off
>
> **Example:**   `national enable packet-ext-seq-mode`

request-reverse-charges
Requests reverse charges for all outgoing calls.

DDN Default   off

GTE Default   on

Example: `national enable request-reverse-charges`

suppress-calling-address
Suppresses the source address in call packets.

DDN Default   off

GTE Default   off

Example: `national enable suppress-calling-addresses`

throughput-class-negotiation
Enables the registration of throughput class.

DDN Default   off

GTE Default   on

Example: `national enable throughput-class-negotiation`

National Disable

Use the **national disable** command to disable a feature defined by the National
Personality configuration.

Syntax: national disable   accept-reverse-charges
    ccitt
    flow-control-negotiation
    frame-ext-seq-mode
    osi-84
    osi-88
    packet-ext-seq-mode
    request-reverse-charges
    suppress-calling-addresses
    throughput-class-negotiation

National Set

Use the **national set** command to set one or all of the default values made to the
National Personality configuration.

Syntax: national set   call-req
    clear-req . . .
    disconnect-procedure . . .
    dp-timer
    frame-window-size
    network-type . . .
    n2-timeouts
    packet-size . . .
    reset . . .
    restart . . .
    min-recall
    min-connect
    collision-timer
    standard-version

<u>t1</u>-timer
<u>t2</u>-timer

call-req
> Specifies the number of ten second intervals permitted before giving up on a
> call request and clearing it. A zero indicates an infinite wait.
>
> **DDN Default**   20 decaseconds
>
> **GTE Default**   20 decaseconds
>
> **Example:**   `national set call-req 20`

clear-req *retries* OR *timer*
> Specifies the number of clear request retransmissions.
>
> *Retries*   Number of clear request transmissions permitted before action is
> taken.
>> **DDN Default**   retries=1
>>
>> **GTE Default**   retries=1
>
> *Timer*   Number of ten second intervals to wait before retransmitting a call
> request packet. A zero in the timer value indicates an indefinite
> wait.
>> **DDN Default**   18 decaseconds
>>
>> **GTE Default**   18 decaseconds
>
> **Example:**   `national set clear-req retries 2`

disconnect-procedure *passive* OR *active*
> Specifies the type of disconnect procedure to use when disconnecting.
>
> **DDN Default**   passive
>
> **GTE Default**   passive
>
> **Example:**   `national set disconnect-procedure active`
>
> *Passive*   Specifies that DISC frames are not used when disconnecting.
>
> *Active*   Specifies that DISC frames are used when disconnecting.

dp-timer
> Specifies the number of milliseconds that the frame level remains in a
> disconnected state. Zero indicates immediate transition from disconnected
> phase to link setup state.
>
> **DDN Default**   500 milliseconds
>
> **GTE Default**   500 milliseconds
>
> **Example:**   `national set dp-timer 300`

frame-window-size
> Specifies the number of frames that can be outstanding before
> acknowledgement.
>
> **DDN Default**   7
>
> **GTE Default**   7
>
> **Example:**   `national set frame-window-size 7`

network-type *CCITT* OR *DDN*
> Specifies the type of network being supported.

> > **DDN Default**   DDN

> > **GTE Default**   CCITT

> > **Example:**   `national set network-type DDN`

> > *CCITT*   Specifies the CCITT convention.
> > *DDN*   Specifies the DDN convention.

n2-timeouts
> Specifies the number of times the retransmit timer (T1) can expire before the interface is recycled.

> > **DDN Default**   20

> > **GTE Default**   20

> > **Example:**   `national set n2-timeouts 10`

packet-size *default* OR *maximum* OR *window*
> Specifies the size of the packet.

> > **Example:**   `national set packet-size maximum 256`

> > *default*   Number of bytes in the data portion of the packet. Possible options include 128, 256, 512, 1024, 2048, and 4096. This value is used in the absence of packet size negotiation. *Default* cannot be greater than *maximum*.

> > > **DDN Default**   128

> > > **GTE Default**   128

> > *maximum*   Maximum number of bytes in the data portion of the packet. Possible options include 128, 256, 512, 1024, 2048, and 4096.

> > > **DDN Default**   256

> > > **GTE Default**   256

> > *window*   Number of outstanding I-frames permitted before acknowledgement is required. The range is determined by the National Personality Packet Modulus.

> > > Related configuration parameters are

> > > - Protocol max default window
> > > - Set default window size

reset *retries* OR *timer*
> Specifies the number of reset request retransmissions.

> > **Example:**   `national set reset retries 2`

> > *retries*   Number of reset request transmissions permitted before the call is cleared. The range is 0 to 255.

> > > **DDN Default**   1

> > > **GTE Default**   1

> > *timer*   Number of ten second intervals to wait before retransmitting a reset request packet. The range is 0 to 255. A zero in the timer value indicates an indefinite wait.

> > **DDN Default** 18 decaseconds
> >
> > **GTE Default** 18 decaseconds

restart *retries* OR *timer*
> Specifies the number of restart request transmissions.
>
> **Example:** `national set restart timer 12`
>
> *retries* Number of restart request transmissions permitted before the interface is recycled. The range is 0 to 255.
>
> > **DDN Default** 1
> >
> > **GTE Default** 1
>
> *timer* Number of ten second intervals to wait before retransmitting a restart request packet. The range is 0 to 255. A zero in the timer value indicates an indefinite wait.
>
> > **DDN Default** 18 decaseconds
> >
> > **GTE Default** 18 decaseconds

min-recall
> Specifies the minimum number of seconds to wait prior to reinitiating a call to open an SVC. The range is 0 - 255 seconds.
>
> **DDN Default** 10 seconds
>
> **GTE Default** 10 seconds
>
> **Example:** `national set min-recall`

min-connect
> Specifies in seconds, the minimum amount a time an SVC will remain established once the connection is made barring any error conditions. The range is 0 - 255 seconds.
>
> **DDN Default** 90 seconds
>
> **GTE Default** 90 seconds
>
> **Example:** `national set min-connect`

collision-timer
> Specifies in seconds, the time delay used prior to reinitiating a call to open an SVC if the original attempt resulted in a call collision. The range is 0 - 255 seconds.
>
> **DDN Default** 10 seconds
>
> **GTE Default** 10 seconds
>
> **Example:** `national set collision-timer`

standard-version
> Options are none, v1980, v1984, and v1988.
>
> **DDN Default** 1984
>
> **GTE Default** 1984
>
> **Example:** `national set standard-version v1984`

t1-timer
> Specifies the frame retransmit time in hundredths of a second. The range is 0.05 - 255.99.

> **DDN Default**   4 seconds
>
> **GTE Default**   4 seconds
>
> **Example:**   `national set t1-timer 3.2`

t2-timer
> Specifies the amount of time in seconds to delay before acknowledging an
> I-frame. This is an optimization parameter. Setting the timer to 0 disables it.
> The range is 0 - 255.99
>
> **DDN Default**   4
>
> **GTE Default**   4
>
> **Example:**   `national set t2-timer 2.5`

National Restore

Use the **national restore** command to restore one or all of the default values made
to the National Personality configuration via the **national set**, **national enable**, or
national disable command. Default values can be found in Table 24-2 on
page 24-3 and Table 24-3 on page 24-3.

Syntax: national restore
- all
- accept-reverse-charges
- call-req
- ccitt
- clear-req . . .
- disconnect-procedure . . .
- dp-timer
- flow-control-negotiation
- frame-ext-seq-mode
- frame-window-size
- network-type . . .
- n2-timeouts
- osi-84
- osi-88
- packet-size . . .
- packet-ext-seq-mode
- request-reverse-charges
- reset . . .
- restart . . .
- standard-version
- suppress-calling-addresses
- throughput-class-negotiation
- t1-timer
- t2-timer

Add

Use the **add** command to add an X.121 address, an DDN X.25 Address, a protocol
configuration, or a PVC definition.

Syntax: add
- address
- htf-address
- protocol
- pvc

Configuring the X.25 Network Interface

address
> Adds an X.121 address translation for a protocol supported in the configuration of the router. The prompts that appear depend on the protocol address that you are adding.   (See the following examples.) The protocol address and X.121 address being entered represent the protocol and X.121 DTE address of the remote DTE connecting to the router X.25 interface. The **set address** command is used to set the local X.25 address.

> **Example: add address**

> IP example:

```
Protocol [IP]?  IP
IP Address [0.0.0.0]? 128.185.1.2
Enc Priority 1 []?
Enc Priority 2 []?
Enc Priority 3 []?
X.25 Address []? 1234590
```

> IPX example:

```
Protocol [IP]?  IPX
CUD Field Usage (Standard or Proprietary)
IPX Host Number (in hex) []?
Enc Priority 1 []?
Enc Priority 2 []?
X.25 Address []?
```

> *Protocol*   Specifies the protocol type of the address mapping you are adding. The valid values are APPN, DECnet, DLSw, IP, IPX and VINES. The default is IP.

> *Enc Priority*
> > Determines the encapsulation type, as defined in RFC 1356, that will be put in the CUD. For IP, valid choices are CC, or SNAP. For IPX, valid choice is SNAP. Enc Priority 1 is used in the first call attempt; if this fails, then Priority 2 is used and so on.

> *IP Address*
> > Specifies the destination's IP address.

> *CUD Field Usage*
> > This field is for IPX to X.25 address mapping only. It determines how the Call User Data (CUD) field is filled in when call request packets are received for IPX. The CUD field can be either Standard or Proprietary. Standard indicates that the usage is protocol multiplexing used in RFC 1356. Proprietary indicates a proprietary CUD field that can only be used with 2216 or compatible routers. The default is Standard.

> *IPX Host Number*
> > Specifies the IPX host number of the destination.

> *X.25 Address*
> > Specifies the X.121 DTE address of the remote DTE connecting to the router X.25 interface. The maximum address length is 15 digits.

htf-address
> Adds a Defense Data Network (DDN) X.25 address translation.

> **Example: add htf-address**

```
Protocol [IP]
Current HTF address
```

Protocol   Specifies the protocol that you are running over the X.25 interface. DDN supports IP only.

Current HTF address
         Specifies the destination X.121 address in Host Table Format (HTF) format. Also see ddn-address-translations in the Enable/Disable commands section.

protocol
    Enables a protocol encapsulation and defines the associated parameters.

Example: **add protocol**

```
Protocol [IP]
Window Size [2]
Default Packet Size [128]
Maximum Packet Size [256]
Circuit Idle Time [30]
Maximum VCs [4]
```

QLLC example:

```
X.25 Config> add prot
Protocol [IP]? dls
Circuit Idle time [20]?
QLLC response timer [20]?
QLLC response count [10]?
Accept Reverse Charges [N]?
Request Reverse Charges [N]?
Station Type (1) PRI (2) SEC (3) PEER [3]?
Window size [128]?
Max message size [256]?
Call User Data (in HEX) [0000000000000000]?
```

Protocol   Specifies which protocol's encapsulation parameters you want to add: APPN, XTP, IP, DECnet, IPX, DLSw, or Banyan VINES. The default is IP.

Window Size   Specifies the maximum negotiable packet window size, the number of packets that can be outstanding before requiring packet confirmation. The default is 2. The window size can be negotiated down to 1 by the called DTE.

         Related configuration parameters are:

          • Set Default Window

Default Packet Size
         Specifies the default requested packet size for SVCs. This value serves as the lowest negotiable packet size and must be equal to or less than the maximum packet size specified with the **national set packet-size** command. The maximum *default packet size* is 4096 bytes. The default value for this parameter is 128 bytes.

         Related configuration parameters are:

          • National Set Packet Size Default

          • National Set Packet Size Maximum

Maximum Packet Size
         Specifies the maximum negotiable packet size for SVCs. This value must be equal to or less than the maximum packet size specified with the **national set packet-size** command. The

default value for this parameter is 128 bytes. The maximum value that may be configured for this parameter is 4096 bytes. This value is utilized in calculating the maximum frame size for this X.25 interface.

Related configuration parameters are:

- National Set Packet Size Default
- National Set Packet Size Maximum

Circuit Idle Time
Specifies the number of seconds that an SVC can be idle before it is cleared by the router. The range is 0 to 65365. The default is 30 seconds. A 0 (zero) specifies that the circuit is never cleared by the router.

Maximum VCs Specifies the maximum number of circuits that are open to the same DTE address for a protocol. Refer to RFC 1356 for information on utilizing this parameter. The Valid range is 1 to 10. The default is 4.

The following are QLLC unique parameters:

QLLC response timer
The number of seconds to wait for a Q-response packet before retransmitting.

QLLC response count
The maximum number of times QLLC will retransmit. Upon exhausting this number of retries, the upper layer is notified which may result in the circuit being cleared or reset by the router.

Accept Reverse Charges
Allows this protocol to override the setting of this National Personality parameter. This does not affect the National Personality parameter.

Request Reverse Charges
Allows this protocol to override the setting of this National Personality parameter. This does not affect the National Personality parameter.

Station Type Specifies the default station type for this protocol:

Pri   Primary Station
Sec   Secondary Station
Peer   Peer Station

Max message size
The maximum message size for this protocol. Specify a value that is less than, or equal to, the Max MTU size of the interface.

Call User Data Specifies the default CUD field used in call packets for this protocol. Specify from 1-to-16 characters. If you do not specify characters, the default 0xC3 is used.

pvc
Adds PVC, window size, and packet size definitions.

Example: `add pvc`

IP example:

```
Protocol [IP]? IP
Packet Channel [1]?
Destination X.25 Address[]?
Window Size [2]?
Packet Size [128]?
```

Protocol   Specifies which protocol's encapsulation you want to modify: APPN, XTP, DECnet, Banyan Vines, DLSw, IP or IPX. The default is IP.

Packet Channel
Specifies the circuit number of the PVC.

Destination X.25 Address
Specifies the X.25 address of the PVC's destination.

Window Size   Specifies the number of packets that can be outstanding before requiring packet confirmation. The default is 2.

Related configuration parameters are:

- Set Default Window

Packet Size   Specifies the maximum negotiable packet size for PVCs. This value must be equal to or less than the maximum packet size specified with the **national set packet-size** command. The default value for this parameter is 128 bytes. The maximum value that may be configured for this parameter is 4096 bytes. This value is utilized in calculating the maximum frame size for this X.25 interface.

Related configuration parameters are:

- Nat Set Packet Size Default

- Nat Set Packet Size Maximum

Change

Use the **change** command to change an X.121 address, an DDN X.25 Address, a protocol configuration, or a PVC definition.

Note: To change an IP address that is associated with an X.121 address, you must delete the record that contains the address correlation, then redefine the address mapping.

Syntax: change address
                    htf-address
                    protocol
                    pvc

address
Modifies a X.121 address translation. The prompts that appear depend on the protocol that is changing.

Example: `change address`

IP example:

```
Protocol [IP]  IP
IP Address [0.0.0.0]?
Enc Priority []?
X.25 Address [00000124040000]?
```

IPX example:

```
Protocol [IP]  IPX
CUD Field Usage (Standard or Proprietary) [Standard]?
IPX Host number (in hex) []?
Enc Priority []?
X.25 Address [00000124040000]?
```

htf address

Changes a Defense Data Network (DDN) X.25 address translation.

Example: `change htf-address`

```
Protocol [IP]
Change HTF address [0.0.0.0]?
New HTF address [10.4.0.124]?
```

protocol

Changes a protocol configuration definition.

Example: `change protocol`

```
Protocol [IP]
Window Size [2]
Default Packet Size [128]
Maximum Packet Size [256]
Circuit Idle Time [30]
Maximum VCs [6]
```

QLLC example:

```
X.25 Config> change prot
Protocol [IP]? dls
Idle Timer [30]?
QLLC response timer (in decaseconds) [15]?
QLLC response count [255]?
Accept Reverse Charges [N]?
Request Reverse Charges [N]?
Station Type (1) PRI (2) SEC (3) PEER [3]?
Max Packet Size [256]?
Packet Window size [7]?
Max message size [2048]?
Call User Data (in HEX, 0 for Null) []? C3010000525450
```

pvc

Changes PVC, window size, and packet size definitions.

Note: To change the protocol, packet channel or destination X.25 address, you must delete the record which contains the definition, then add it back with the changed parameters.

Example: `change pvc`

IP example:

```
Protocol [IP]? IP
Packet Channel [1]?
Destination X.25 Address []?
Window Size [2]?
Packet Size [128]?
```

Delete

Use the **delete** command to delete an X.121 address, a protocol configuration definition, or a PVC definition.

Syntax: <u>d</u>elete   <u>a</u>ddress
                   <u>p</u>rotocol . . .
                   <u>p</u>vc

address
    Deletes an X.121 address translation.

    Example: **delete address**

    IP example:

```
Protocol [IP]?
IP Address [0.0.0.0]?
```

    IPX example:

```
Protocol [IP]? IPX
IPX Host Number (in hex) [2]?
```

protocol *prot-type*
    Deletes a protocol encapsulation configuration definition. *Prot-type* is the name or number of the protocol encapsulation that is currently defined in the router's configuration.

    Example: **delete protocol IPX**

pvc
    Deletes a PVC definition.

    Example: **delete pvc**

```
Protocol [IP]?
Destination X.25 Address []?
```

List

Use the **list** command to display the current configuration for the specified parameter.

Syntax: <u>l</u>ist   <u>a</u>ddress
           <u>a</u>ll
           <u>d</u>etailed
           <u>pro</u>tocols
           <u>p</u>vc
           <u>s</u>ummary

address
    Lists all the X.121 address translations.

    Example: **list address**

```
IF#     Prot #     Active Enc    Protocol ->    X.25 address
1       0(IP)      CC            10.1.2.3 ->    1238765742
1       7(IPX)     SNAP          10       ->    12389
```

all
    Lists all the X.25 addresses, National Personality parameters, all defined protocols and their values, and all defined PVCs.

    Example: **list all**

```
X.25 Configuration Summary

Node Address:      313131
Max Calls Out:         4
Inter-Frame Delay:     0     Encoding:  NRZ
Speed:             64000     Clocking:  Internal
MTU:                2048     Cable:     V.35 DCE
Lower DTR:   Disabled
Default Window:    2         SVC idle:  30 seconds
National Personality: GTE Telenet (DTE)
PVC            low: 1    high: 1
Inbound        low: 0    high: 0
Two-Way        low: 2    high: 64
Outbound       low: 0    high: 0
Throughput Class in bps Inbound:  2400
Throughput Class in bps Outbound: 2400


X.25 National Personality Configuration

Follow CCITT: on    OSI 1984:    on        OSI 1988: off
Request Reverse Charges:  on  Accept Reverse Charges:    on
Frame Extended seq mode:  off Packet Extended seq mode:   off
Incoming Calls Barred:    off Outgoing Calls Barred:      off
Throughput Negotiation:   on  Flow Control Negotiation:   on
Suppress Calling Addresses: off
DDN Address Translation:  off
Call Request Timer:     20 decaseconds
Clear Request Timer:    18 decaseconds (1 retries)
Reset Request Timer:    18 decaseconds (1 retries)
Restart Request Timer:  18 decaseconds (1 retries)
Min Recall Timer        10 seconds
Min Connect Timer       90 seconds
Collision Timer          5 seconds
T1 Timer:  4.00 seconds    N2 timeouts: 20
T2 Timer:  2.00 seconds     DP Timer:    500 milliseconds
Standard Version:      1984      Network Type: CCITT
Disconnect Procedure: passive
Window Size    Frame:  7     Packet: 2
Packet Size    Default: 128   Maximum: 256


X.25 protocol configuration

No protocols defined

X.25 PVC configuration

No PVCs defined

X.25 address translation configuration
No address translations defined
```

detailed

Lists the value of all the default parameters that the **national set** command modifies. Descriptions of the screen display are listed in the **national set** command described later in this chapter.

Example: `list detailed`

```
X.25 National Personality Configuration

Follow CCITT: on            OSI 1984: on    OSI 1988: off
Request Reverse Charges: on   Accept Reverse Charges:    on
Frame Extended seq mode: off  Packet Extended seq mode:   off
Incoming Calls Barred:   off  Outgoing Calls Barred:      off
Throughput Negotiation:  on   Flow Control Negotiation:   on
Suppress Calling Addresses: off
DDN Address Translation: off
Call Request Timer:     20 decaseconds
Clear Request Timer:    18 decaseconds (1 retries)
Reset Request Timer:    18 decaseconds (1 retries)
```

```
Restart Request Timer:  18 decaseconds (1 retries)
Min Recall Timer       10 seconds
Min Connect Timer      90 seconds
Collision Timer         5 seconds
T1 Timer: 4.00 seconds     N2 timeouts: 20
T2 Timer: 2.00 seconds   DP Timer:    500 milliseconds
Standard Version:    1984      Network Type: CCITT
Disconnect Procedure: passive
Window Size    Frame:  7       Packet: 2
Packet Size    Default: 128   Maximum: 256
```

protocols

Lists all the defined protocol configurations. See "Add" on page 24-16 for a description of the parameters.

Example: `list protocols`

```
X.25 protocol configuration

Protocol      Window          Packet-Size        Idle    Max
 Number       Size          Default Maximum    Time    VCs

 0(IP)          2             128   256          30      4

QLLC Protocols

Protocol     Packet        Idle      Response    Reverse_Charges   Max       Station
 Number   Window MaxSize  Time    Timer Count  Accept Request   Message      Type

26(DLSW)    7     256      30     15    255      N      N        2048        PEER
          CUD : [C3 01 00 00 52 54 50 ]
```

pvc

Lists all the defined PVCs.

Example: `list pvc`

```
X.25 PVC configuration

Prtcl    X.25 Address    Active Enc    Window    Pkt_len   Pkt_chan
0         8383838383        CC           4        1024       3
```

summary

Lists all the values established by the **set** and **enable** commands. These values modify the X.25 configuration.

Example: `list summary`

```
X.25 Configuration Summary

Node Address:        313131
Max Calls Out:          4
Inter-Frame Delay:      0     Encoding:  NRZ
Speed:         64000          Clocking:  Internal
MTU:            2048          Cable:     V.35 DCE
Lower DTR:   Disabled
Default Window:    2          SVC idle:  30 seconds
National Personality: GTE Telenet (DTE)
PVC           low: 1    high: 1
Inbound       low: 0    high: 0
Two-Way       low: 2    high: 64
Outbound      low: 0    high: 0
Throughput Class in bps Inbound:  2400
Throughput Class in bps Outbound: 2400
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: exit

Example: `exit`

Chapter 25. Monitoring the X.25 Network Interface

This chapter describes the X.25 console commands and includes the following sections:

- "Accessing the Interface Console Process"
- "X.25 Console Commands"
- "X.25 Network Interfaces and the GWCON Interface Command" on page 25-6

Accessing the Interface Console Process

To monitor information related to the X.25 network interface, access the interface console process by doing the following:

1. At the OPCON prompt, enter **talk 5**. For example:

   ```
   * talk 5
   +
   ```

   The GWCON prompt (+) is displayed on the console. If the prompt does not appear when you first enter GWCON, press **Return** again.

2. At the GWCON prompt, enter the **configuration** command to see the protocols and networks for which the router is configured. For example:

   ```
   + configuration
   ```

   See page 6-4 for sample output of the **configuration** command.

3. Enter the **network** command and the number of the X.25 interface.

   ```
   + network 2
   X.25>
   ```

   The X.25 console prompt is displayed on the console. You can then view information about the X.25 interface by entering the X.25 console commands.

X.25 Console Commands

This section summarizes and explains all the X.25 console commands. The X.25 console commands allow you to view the parameters and statistics of the interfaces and networks that transmit X.25 packets. Console commands display configuration values for the physical, frame, and packet levels. You also have the option of viewing the values for all three protocol levels at once.

Enter the X.25 console commands at the X.25> prompt. Table 25-1 on page 25-2 shows the commands.

Table 25-1. X.25 Console Command Summary

Console Command	Function
? (Help)	Lists all the X.25 console commands or lists the options associated with specific commands.
List	Lists individual PVC or SVC statistics and general information.
Parameters	Displays the current parameters for any level of the X.25 configuration.
Statistics	Displays the current statistics for any level of the X.25 configuration.
Exit	Exits the X.25 console process and returns to the GWCON process.

? (Help)

Use the *? (help)* command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
List
Parameters
Statistics
Exit
```

List

Use the **list** command to display the current active PVCs and SVCs.

Syntax: list   pvcs
                       svcs

pvc
  Displays the configured permanent virtual circuits.

  Example: `list pvcs`

svc
  Displays the active switched virtual circuits.

  Example: `list svcs`

```
HCN/  Destination  Originate Transmits Protocol    Totals
State       Address    Call   Queued  Encapsulated Xmts Rcvs Resets

13 D   898280077113   YES    0            IP       8943  261    1
20 D   898280077114   NO     0            IP        943   43    0
42 P   898280077116   YES    6            IP          0    0    0
23 C   898280077117   YES    0            IP       3054  110    0


D  - Data Transfer     P  - Call Progressing
C  - Call Clearing
```

Parameters

Use the **parameters** command to display the current parameters for any level of the X.25 configuration.

Syntax: parameters   <u>all</u>
                    <u>f</u>rame
                    <u>pa</u>cket
                    <u>ph</u>ysical

all

    Displays the parameters for the packet, frame, and physical levels.

    Example: `parameters all`

frame

    Displays the parameters for the frame level.

    Example: `parameters frame`

```
Frame Layer Parameters:
Maximum Frame Size  = 262 Maximum Window Size =    7
Protocol Enabled    = YES Equipment Type      =  DTE
T1 Retransmit Timer =   4 T2 Acknowledge Timer =   2
N2 Retry Counter    =  20 Disconnect Procedure =  PASSIVE
Disconnect Timer    = 500 Network Type        =  GTE
Protocol Options: MOD 128
                    A/B Addressing  YES  Enable SARM  NO
```

packet

    Displays the parameters for the packet level.

    Example: `parameters packet`

```
Packet Layer Parameters:
Default Packet Size = 128  Maximum Packet Size =   256
Log 2 Packet size   =   2  Acknowledge Delay   =     0
Layer Enabled       = YES Default Window Size =     2
Lowest SVC          =   1  Highest SVC         =    64
Lowest PVC          =   0  Highest PVC         =     0
                           T21 (Call)          =    20
T20 (Restart)       = 18  R20 (Retry)          =     1
T22 (Reset)         = 18  R22 (Retry)          =     1
T23 (Clear)         = 18  R23 (Retry)          =     1
Network Type        = GTE  Equipment Type      =   DTE
Recall Timer        = 10 seconds
Min Connect         = 90 seconds
Collision           = 5 seconds
```

physical

    Displays the parameters for the physical level.

    Example: `parameters physical`

```
Physical Layer Parameters:
Interface Type      =   V.35

Maximum Frame Size  =   264  InterFrame Delay  =        2
Configured Speed    =     0  Clocking          = External
Encoding            =   NRZ
Protocol Enabled    =   Yes
```

Statistics

Use the **statistics** command to display the current statistics of any level of the X.25 configuration.

Syntax: <u>s</u>tatistics   <u>a</u>ll
                                    <u>f</u>rame
                                    <u>pa</u>cket
                                    <u>ph</u>ysical

all

Displays the statistics for the packet, frame, and physical levels.

Example: `statistics all`

frame

Displays the statistics for the frame level.

Example: `statistics frame`

```
Frame Layer Counters:      Received          Transmitted
Information Frames          0                     0
RR Command                  0                     0
RR Response                 0                     0
RNR Command                 0                     0
RNR Response                0                     0
REJ Command                 0                     0
REJ Response                0                     0
SABM                        0                    71
SABME                       0                     0
UA                          0                     0
DISC                        0                     0
DM                          0                     0
FRMR                        0                     0
Total Bytes                 0                     0
T1 Timeouts   0   T2 Timeouts        0  N2 Timeouts   1
Bad Address   0   Unsolicited F-Bit  0  Invalid Ctl   0
Frame Layer Miscellaneous:
Queued Output Frames = 0 Protocol Layer State = Link Setup
Send Sequence N(S)   = 0 Receive Sequence N(R)= 0
```

packet

Displays the statistics for the packet level.

Example: `statistics packet`

```
Packet Counters:           Received          Transmitted
Call Request                0                     0
Call Accepted               0                     0
Clear Request               0                     0
Clear Confirm               0                     0
Interrupt Request           0                     0
Interrupt Confirm           0                     0
RR Packet                   0                     0
RNR Packet                  0                     0
REJ Packet                  0                     0
Reset Request               0                     0
Reset Confirm               0                     0
Restart Request             0                     0
Restart Confirm             0                     0
Diagnostic                  0                     0
Data Packet                 0                     0
Data Bytes                  0                     0
Buffers Queued              0                     0
Invalid Packets Received =  0
Switched Circuits Opened =  0
```

physical
> Displays the statistics for the physical level.

> **Example:** `statistics physical`

```
X.25 Physical Layer Counters:
 Rx Bytes                      0   Tx Bytes                   0

 Adapter cable:           V.35 DTE

  Nicknames:    RTS CTS DSR DTR DCD
  PUB 41450:    CA  CB  CC  CD  CF
  State:        ON  ON  ON  ON  ON

 Line speed:             unknown
 Last port reset:        12 minutes, 21 seconds ago

 Input frame errors:
 CRC error                     0   alignment (byte length)      0
 missed frame                  0   too long (>     0 bytes)     0
 aborted frame                 0   DMA/FIFO overrun             0
 L & F bits not set            0
Output frame counters:
 DMA/FIFO underrun errors      0   Output aborts sent          0
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: `exit`

X.25 Network Interfaces and the GWCON Interface Command

While X.25 interfaces have their own console processes for monitoring purposes, the router also displays complete statistics for installed network interfaces when you use the **interface** command from the GWCON environment. (For more information on the **interface** command, refer to Chapter 6, "The GWCON (Monitoring) Process and Commands").

Statistics Displayed for X.25 Interfaces

The following statistics display when you run the **interface** command from the GWCON environment for X.25 interfaces:

```
+interface 11
                                          Self-Test  Self-Test  Maintenance
Nt Nt'  Interface  Slot-Port              Passed     Failed     Failed
11 11   X25/0       Slot: 8  Port: 1         1          0          0

  X.25 MAC/data-link on V.35/V.36 interface
Interface State:  DCD  CTS    Packet Layer    Frame Layer
                  ON   ON         UP              UP

Packet Counters:           Received           Transmitted
 Data Packet                  0                    353
 Data Bytes                   0                  18888
 Buffers Queued               0                      0
 Invalid Packets Received =   0
 Switched Circuits Opened =   0

Frame Layer Counters:      Received           Transmitted
  Information Frames          354                   354

X.25 Physical Layer Counters:
  Rx Bytes                  3316    Tx Bytes            22204

  Adapter cable:              V.35 DTE

  V.24 circuit: 105 106 107 108 109
  Nicknames:    RTS CTS DSR DTR DCD
  PUB 41450:    CA  CB  CC  CD  CF
  State:        ON  ON  ON  ON  ON

  Line speed:             64.000 Kbps
  Last port reset:        1 hour, 20 minutes, 25 seconds ago

  Input frame errors:
   CRC error                     0    alignment (byte length)        0
   missed frame                  0    too long (> 2057 bytes)        0
   aborted frame                 0    DMA/FIFO overrun               0
  Output frame counters:
   DMA/FIFO underrun errors      0    Output aborts sent             0
Interface buffer pool: Total = 57, Free = 56
```

The following table describes the interface statistics:

Nt	Global interface number
Nt '	Reserved for future dial circuit use
Interface	Interface name and number (within interfaces of the same type)
Slot	Slot number of interface
Port	Port number of interface

Self-Test Passed	Number of times self-test succeeded
Self-Test Failed	Number of times self-test failed
Maintenance Failed	Number of maintenance failures
Interface state	Display the current state of the input modem control signals, the packet layer (X.25 layer 3), and the frame layer (X.25 layer 2).
Packet Counters	Provides statistics on packets received and transmitted.
Data Packets	Displays the number of data packets the interface transmits receives on the network
Data Bytes	Displays the number of data bytes the interface transmits receives on the network.
Buffers Queued	Displays the number of buffers currently queued for transmission over the network. These may be frame or packet layer supervisory messages as well as forwarder packets.
Invalid Packets Received	Displays the number of invalid X.25 packets received from the network.
Switched Circuits Open	Displays the number of switched circuits currently open.
Frame Layer Counters	Provides statistics generated from Frame Layer counters.
Information Frames	Displays the number of X.25 Information frames the interface has transmitted and received.
X.25 Physical Layer Counters	Provides statistics generated from Physical Layer counters.
RX Bytes	Display the number of bytes received by the Physical layer.
TX Bytes	Displays the number of bytes transmitted by the Physical layer.
Line speed	The transmit clock rate.
Last port reset	The length of time since the last port reset.
Input frame errors:	
CRC error	The number of packets received that contained checksum errors and as a result were discarded.
Alignment	The number of packets received that were not an even multiple of 8 bits in length and as a result were discarded.
Too short	The number of packets that were less than 2 bytes in length and as a result were discarded.
Too long	The number of packets that were greater than the configured size, and as a result were discarded.
Aborted frame	The number of packets received that were aborted by the sender or a line error.
DMA/FIFO overrun	The number of times the serial interface card could not send data fast enough to the system packet buffer memory to receive them from the network.

Missed frame	When a frame arrives at the device and there is no buffer available, the hardware drops the frame and increments the missed frame counter.
L & F bits not set	On serial interfaces, the hardware sets input-descriptor information for arriving frames. If the buffer can accept the complete frame upon arrival, the hardware sets both the last and first bits of the frame, indicating that the buffer accepted the complete frame. If either of the bits is not set, the packet is dropped, the `L & F bits not set` counter is incremented, and the buffer is cleared for reuse.
	Note: It is unlikely that the `L & F bits not set` counter will be affected by traffic.
Output frame counters:	
DMA/FIFO underrun errors	The number of times the serial interface card could not retrieve data fast enough from the system packet buffer memory to transmit them onto the network.
Output aborts sent	The number of transmissions that were aborted as requested by upper-level software.

Chapter 26. Using and Configuring Frame Relay Interfaces

This chapter describes the Frame Relay configuration commands and includes the following sections:

Frame Relay Overview

The Frame Relay (FR) protocol is a method of transmitting internetworking packets by combining the packet switching and port sharing of X.25 with the high speed and low delay of time division multiplexing (TDM) circuit switching. FR allows you to connect multiple LANs to a single high-speed (1.54 Mbps) WAN link with multiple point-to-point permanent virtual circuits (PVCs). FR offers the following features:

- *High throughput and low delay* Utilizing the *core aspects* (error detection, addressing, and synchronization) of the Link Access Protocol, D-Channel (LAPD) datalink protocol, FR eliminates all network layer (Layer 3) processing. By using only the core aspects, FR reduces the delay of processing each frame.

- *Congestion detection.* Upon receiving Backward Explicit Congestion Notification (BECN), the router initiates a controlled slowdown of traffic, thereby avoiding a complete FR network shutdown.

- *Circuit access and control* As the router dynamically learns about the availability of non-configured circuits (orphan circuits), you can control access to those new circuits.

- *Network management option* As your network requires, the FR protocol can operate with or without a local network management interface.

- *Multiplexing protocols* Using one PVC to pass multiple protocols.

FR provides no error correction or retransmission function. To provide error free end-to-end transmission of data, FR relies on the intelligence of the host devices.

Frame-Relay Network

The FR network consists of the FR backbone (consisting of FR switches provided by the FR carrier) providing the FR service. The router functions as the FR connection device. The router encapsulates FR frames and routes them through the network based on a Data Link Connection Identifier (DLCI). The DLCI is the Medium Access Control (MAC) address that identifies the PVC between the router and the FR destination device. For example, in Figure 26-1, a packet destined to go from router B to router D would have a DLCI of 19 to reach router D; however, a packet destined to go from router D to router B would have a DLCI of 16.
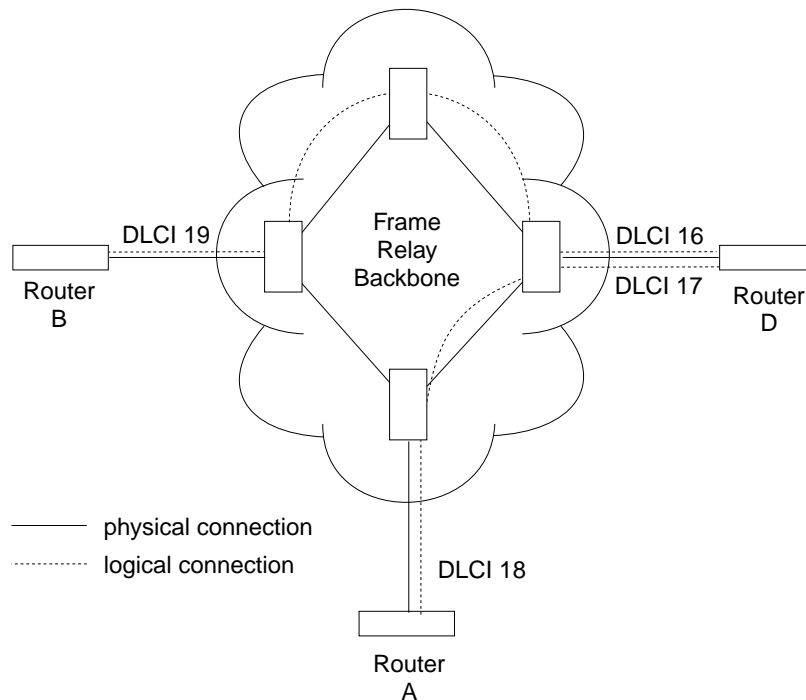
Figure 26-1. DLCIs in Frame Relay Network

A DLCI can have either local or global significance. Local DLCIs are significant at the point of entry to the network, but global DLCIs are significant throughout the network. To the user, however, the DLCI that the router uses to route a packet is the DLCI that the user associates with the frame's global or local destination. DLCIs are configured through the FR configuration process or learned through FR management.

A Frame Relay network has the following characteristics:

- Transports frames transparently The network can modify only the DLCI, congestion bits, and frame check sequence. High-Level Data Link Control (HDLC) flags and zero bit insertion provide frame delimiting, alignment, and transparency.

- Detects transmission, format, and operational errors (frames with an unknown DLCI)

- Preserves the ordering of frame transfer on individual PVCs

- Does not acknowledge or retransmit frames

Frame Relay Interface Initialization

If a Local Management Interface (LMI) is enabled, the FR interface is active when a successful exchange of LMI frames occurs between the router and the FR switch; however, no data can be received from or transmitted to another router until an LMI status message indicates that the PVC status for the DLCI to the other router is active. Also, there are instances where the FR interface state is tied to PVC states and the interface doesn't come up even if LMI exchanges are successfully occurring (for additional information, see "Configuring PVC States to Affect the Frame Relay Interface State" on page 26-4).

PVC status appears for all PVCs as either active or inactive. An active PVC has a completed connection to an end system. An inactive PVC does not have a completed connection to an end system because either an end system or an FR switch is off-line.

For example, in Figure 26-2 router B has a configured PVC to router D. Router B is successfully interacting with FR management through FR switch B. Because either another FR switch is down or the end system is down, the end-to-end PVC connection is not established. Router B receives an inactive status for that PVC.
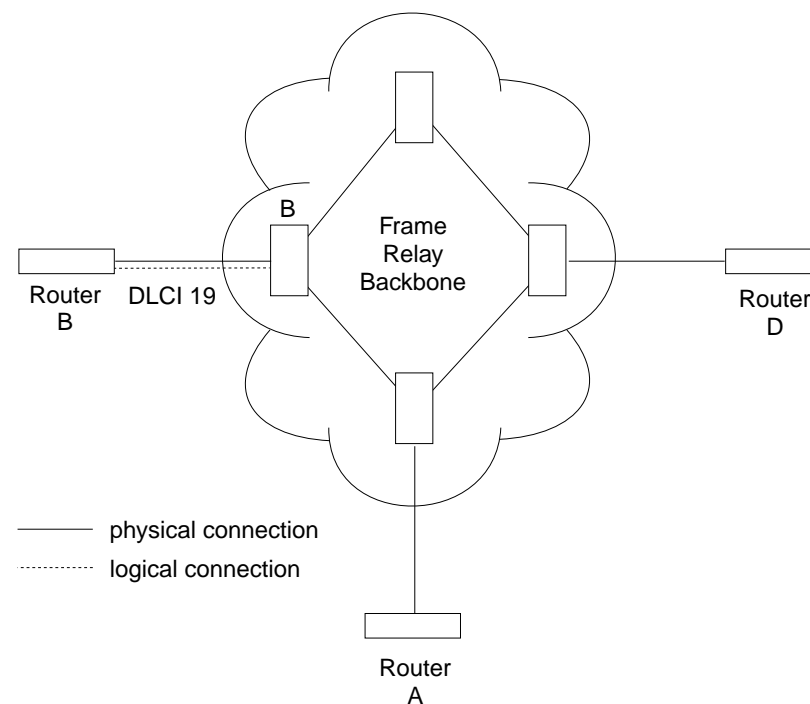
Figure 26-2. DLCIs in Frame Relay Network

When the Local Managment Interface (LMI) is disabled, the FR interface is running on a serial line and a DTE cable is being used, the FR protocol asserts the DTR and RTS modem control signals. (The Control signal is asserted for X.21). The FR interface goes up once the DSR, CTS, and DCD modem control signals are on. (When X.21 is used, the FR interface goes up once the Indication modem control signal is on.) The FR interface is down or in the testing state if either DSR, CTS, or DCD are off or, when X.21 is used, the Indication signal is off. Therefore, you need to ensure that the modem, modem eliminator, or DSU that is used drops one or more of these signals when the physical connection to the FR switch or the other FR DTE (if configured for FR DTE to DTE connectivity) is lost.

Orphan Circuits

An *orphan circuit* is any PVC that is not configured for your router but is learned indirectly through the actions of the network management entity. For example, Figure 26-3 assumes that router B has a configured PVC to router D, but none to router A. Router A configures a PVC to router B. Router B would then learn about the PVC to router A from LMI messages and classify it as an orphan.

Orphan circuits are treated the same as configured circuits except that you may enable or disable their use with the **enable orphan-circuit** and **disable orphan-circuit** commands.

By disabling orphan circuits, you add a measure of security to your network by preventing any unauthorized entry into your network from a non-configured circuit. By enabling orphan circuits, you allow the router to forward packets over circuits you did not configure. Packets that would normally be dropped are now forwarded.
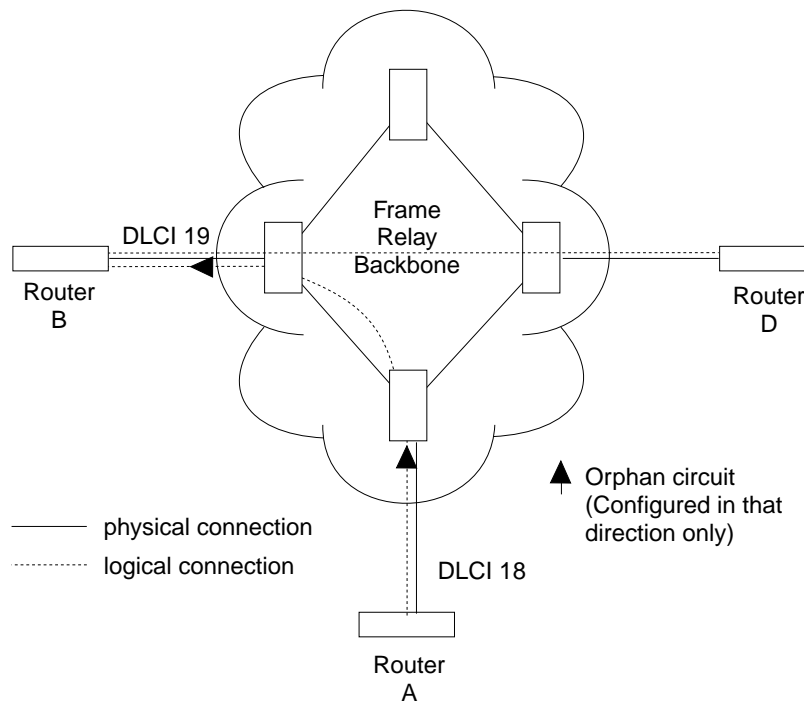
Figure 26-3. Orphan Circuit

Configuring PVC States to Affect the Frame Relay Interface State

You can control the operation of your Frame Relay interface by

1. Enabling the "No-PVC" feature or
2. Configuring "required PVCs" or
3. Configuring "required PVC groups."

By enabling the Frame Relay "No-PVC" feature, the Frame Relay interface becomes inactive when there are no active PVCs on the interface. If at least one PVC is active, the Frame Relay interface becomes active when a successful LMI exchange occurs between the router and the FR switch.

You can configure a PVC as a "required PVC." If a PVC is required but not in a group, the Frame Relay interface becomes inactive when the PVC becomes inactive. When the PVC becomes active, the interface is activated following a successful exchange of LMI frames between the router and the Frame Relay switch.

If multiple PVCs are required and are not in a PVC group, the interface is not activated until all required PVCs are active.

If a required PVC belongs to a PVC group, the Frame Relay interface becomes inactive when all PVCs in the PVC group are inactive. If at least one PVC in the group is active, the interface becomes active following a successful exchange of LMI frames between the router and the FR switch. If there are multiple PVC groups, the interface does not become active until at least one PVC *in each group* is active.

A "required PVC group" is a group of circuits associated by name, where "name" is the name of the required PVC group.

These features can be used with WAN Reroute so that an alternate link can be brought up if all PVCs, required PVCs, or a group of PVCs become inactive on the primary FR link.

Frame Relay Frame

An FR frame consists of a fixed size address field with variable sized encapsulated user data. Figure 26-4 illustrates a Frame-Relay frame format.

```
        8     7     6     5     4    3    2    1
Octet  ┌─────────────────────────────────────────┐
  1    │          HDLC Flag = 0x7e                │
  2    │    Data Link MSB/LSB (DL)    │ C/R │ EA │
  3    │ Connection ID (CI)    │FECN│BECN│ DE │ EA │
       │                                          │
   .   │              user                      . │
   .   │              data                      . │
   .   │                                        . │
       │              Frame Check                 │
       │              Sequence                    │
  N    │          HDLC Flag = 0x7e                │
       └─────────────────────────────────────────┘
```

Figure 26-4. Frame-Relay Frame Format

HDLC Flags

Located in the first and last octet, these flags indicate the beginning and end of the frame.

Data Link Connection Identifier (DLCI)

This 10-bit routing ID resides in bits 3-8 of octet 2 and bits 5-8 of octet three. The DLCI is the MAC address of the circuit. The DLCI allows the user and network management to identify the frame as being from a particular PVC. The DLCI enables multiplexing of several PVCs over one physical link.

Command/Response (C/R)

This field's use is not defined within the Frame-Relay standards and the field is passed transparently across the network.

Extended Address

This version of FR does not support extended addressing.

Forward Explicit Congestion Notification (FECN)

The FR backbone network sets this bit to 1 to notify the user receiving the frame that congestion is occurring for the PVC in the direction the frame is being sent.

APPN High Performance Routing (HPR) uses detection of this bit set to allow Rapid Transport Protocol's adaptive rate-based flow and congestion control algorithm to adjust the data send rate. This alogrithm prevents traffic bursts and congestion, maintaining a high level of throughput.

Backward Explicit Congestion Notification (BECN)

The FR backbone network sets this bit to 1 to notify the user that the frames sent by this router for this PVC have encountered congestion. The router then initiates a *throttle down* to a rate equal to or less than the user-defined CIR when CIR or congestion monitoring are enabled. The CIR for a PVC is supplied by the FR service provider and is configured using the **add permanent-virtual-circuit** command.

Discard Eligibility (DE)

The network may discard transmitted data exceeding the CIR on a PVC. The DE bit is set by the end station to indicate discard eligibility. This version of FR does not set the DE bit, but does log an event if it receives a frame with the DE bit set.

User Data

This field contains the protocol packet being transmitted. This field can contain a maximum of 8188 octets; however, the frame check sequence (FCS) can effectively detect errors only on a maximum of 4096 octets of data. The protocol data is preceded by a Frame Relay encapsulation header as defined in RFC 1490.

Frame Check Sequence

This field is the standard 16-bit cyclic redundancy check (CRC) that HDLC and LAPD frames use. This field detects bit errors occurring in the bits of the frame between the opening flag and FCS.

Frame Forwarding over the Frame Relay Network

When the FR protocol receives a packet for encapsulation, it compares the packet's network address to the entries in the Address resolution Protocol (ARP) cache. If the ARP cache contains the DLCI number that matches the network address, the FR protocol encapsulates that packet into a frame and transmits the frame over its specified local DLCI. If the ARP cache does not contain a match, the FR protocol sends out an ARP request over all configured PVCs on the interface. When the appropriate end-point responds with an ARP response, the FR protocol adds its local DLCI that received the ARP response to the ARP cache. Subsequent data packets directed to the same network address are then encapsulated into a frame and sent out over its local DLCI.

Protocol Addresses

Protocol addresses can be either mapped statically to FR network PVC addresses or discovered dynamically through Inverse ARP or ARP. (For more information on ARP and Inverse ARP, see the *Protocol Configuration and Monitoring Reference*.) Either method is protocol-dependent as illustrated in Table 26-1.

Note: Static protocol addresses are also referred to as static ARP entries. A static ARP entry is added to the configuration with the **add protocol-address** command.

Table 26-1. Protocol Address Mapping

Protocol Type	ARP and Inverse ARP Usage	Static Mapping	PVC Configured at Protocol Configuration
AP2	Yes	Yes	No
IP	Yes	Yes	No
IPX	Yes	Yes	No
Banyan VINES	No	No	No
DNA IV	Yes	Yes	No
OSI*	No	No	Yes

* You must configure OSI at the protocol level to map the protocol address to the FR PVC.

Multicast Emulation and Protocol Broadcast

Multicast emulation is an optional feature that allows protocols requiring multicast such as ARP to function properly over the FR interface. With multicast emulation, a multicast frame is transmitted on each active PVC. By using the **enable** and **disable multicast** commands, you can turn this feature on or off. Protocols that utilize multicast are AP2, ARP, Banyan VINES, DNA4, IP, and IPX.

Protocol broadcast is another optional feature that allows the IP RIP protocol to function properly over the FR interface. By using the **enable protocol-broadcast** and **disable protocol-broadcast** commands, you can turn this feature on or off.

Frame Relay Network Management

The supplier of the FR network backbone provides FR network management. It is management's responsibility to provide FR end-stations (routers) with status and configuration information concerning PVCs available at the interface.

The FR protocol supports the ANSI T1.617 Annex D, ITU-T Q.933 Annex A (also referred to as CCITT Q.933 Annex A), and the Interim Local Management Interface (LMI) management entities. You can turn these entities on or off using the **enable** and **disable** LMI configuration commands. Specifically, FR network management provides the following information:

- Notification of additional PVCs (orphans) and whether they are active or inactive, or notification of any PVC deletions.

- Notification of the availability of a configured PVC. The availability of a PVC is indirectly related to the successful participation of the PVC end-point in the *heartbeat polling* process, which is detailed in "Link Integrity Verification Report" on page 26-9.

- Verification of the integrity of the physical link between the end-station and network by using a *keep alive* sequence number interchange.

Although the FR interface supports network management, it is not necessary for management to run on the FR backbone for the interface to operate over the FR backbone. For example, you may want to disable management for back-to-back testing.

Management Status Reporting

Upon request, FR management provides two types of status reports, a full status report and a link integrity verification report. A full status report provides information about all PVCs the interface knows about. A link integrity verification report verifies the connection between a specific end station and a network switch. All status inquiries and responses are sent over DLCI 0 for ANSI T1.617 Annex D and ITU-T Q.933 Annex A, or DLCI 1023 for interim LMI management.

Full Status Report

When the FR interface requires a full status report, the router's FR protocol sends a status enquiry message to the FR network backbone requesting a full status report. A status enquiry message is a request for the status of all PVCs on the interface. Upon receiving this request, FR management must respond with a full status report consisting of the link integrity verification element and a PVC status information element for each PVC. (See "Link Integrity Verification Report" on page 26-9.)

The PVC status information element contains the following information: the local DLCI number for the particular PVC; the state of the PVC (active or inactive); and whether the PVC is new or an existing PVC that management already knows about.

Note: The number of PVCs supplied at the FR interface is restricted by the network frame size and the amount of individual PVC information elements that can fit into a full status report. For example, 202 is the maximum number of PVCs for a network with a 1K frame size.

Link Integrity Verification Report

The link integrity verification report, sometimes referred to as *heartbeat polling*, contains the link integrity verification element. This element is where the exchange of the send and receive sequence numbers takes place. By exchanging sequence numbers, management and the end station can evaluate the integrity of the synchronous link. The send sequence number is the current send sequence number of the message originator. The receiver looks at this number and compares it to the last send sequence number to verify that this number is incrementally correct. The receive sequence number is the last send sequence number that the originator sent out over the interface. It is the receiver's responsibility to place a copy of the send sequence number into the receive sequence number field. This way the originator can ensure that the receiver receives and interprets the frames correctly.

When an end-station fails to participate in this polling process, all remote end-stations with logically attached PVCs are notified through management's full status report mechanism that the PVC is inactive.

Frame Relay Data Rates

This section introduces data rates for Frame Relay permanent virtual circuits (PVCs).

Committed Information Rate (CIR)

The CIR is the data rate that the network commits to support for the PVC under normal, uncongested conditions. Any PVC that is configured or is learned is provided a CIR (by the FR service provider). The CIR is a portion of the total bandwidth of the physical link of either 0 or between 300 bps and 2M reserved for the PVC. 64 Kbps or a single DS0 channel is most common. You define the CIR with the **add permanent-virtual-circuit** or the **change permanent-virtual-circuit** configuration command. You can also dynamically change the CIR with the **set circuit** console command.

Some Frame Relay switches allow a value of 0 to be configured for CIR. When CIR is equal to 0, little or no bandwidth is reserved in the Frame Relay network backbone for the PVC, and the PVC's traffic uses non-reserved bandwidth.

Orphan Circuit CIR

The router assigns orphan circuits a CIR of 64 Kilobits per second (Kbps). If you are relying on the orphan circuit to route important data, it is recommended that you add a PVC in place of the orphan. Doing this, you can assign a CIR that the network commits to support.

Committed Burst (Bc) Size

The *committed burst (Bc) size* is the maximum amount of data (in bits) that the network commits to deliver during a *calculated time (Tc) interval* . The Tc is equal to the Bc divided by the CIR (Tc = Bc / CIR). If you configure 0 for CIR, Tc is equal to Be divided by the line speed (or access rate (AR)) (Tc = Be/AR).

For example, if you set a PVC's CIR to 9600 bps and the committed burst size to 14 400 bits, the time period is 1.5 sec. (14 400 bits / 9600 bps = 1.5 sec). This

means that the PVC is allowed to transmit a maximum of 14 400 bits in 1.5 seconds.

This parameter is important because of the relationship between the committed burst size and the maximum frame size. If the maximum frame size in bits is greater than the committed burst size, the network may discard frames whose size exceeds the committed burst size. Therefore, the committed burst size should be greater than or equal to the maximum frame size. It should also equal the burst size set up with the network provider.

Use the **add permanent-virtual-circuit** and **change permanent-virtual-circuit** configuration commands to set the committed burst size. The **set circuit** console command can be used to dynamically change the committed burst size.

The router assigns orphan circuits a committed burst size of 64 Kilobits. If you configure 0 for CIR, then the committed burst (Bc) size also equals 0.

Excess Burst (Be) Size

The *excess burst (Be) size* is the maximum amount of uncommitted data the router can transmit on a PVC in excess of the Bc during the Tc (Tc = Bc / CIR) when CIR and Bc are nonzero. When CIR = 0, Tc = Be / AR where the access rate (AR) is the line speed.

The network delivers this excess data with a lower probability of success than committed burst size data. Set the Be to a value greater than zero only if you are willing to accept the risk of discarded data and its effect on higher-layer protocol performance. The Be should equal the value set up with the network provider.

Use the **add permanent-virtual-circuit** command or the **change permanent-virtual-circuit** command during frame-relay configuration to set the excess burst size. You can also use the **set circuit** console command to dynamically change the excess burst size. A default of 0 applies to orphan circuits. If you configure 0 for CIR, then you must configure a non-zero value for the excess burst (Be) size.

Line Speed

The *line speed* is the interface's line speed.

The FR interface's line speed is configured using the **set line-speed** configuration command. The line speed must be configured when internal clocking is used. However, it is recommended that you configure a line speed for external clocking since the router uses the line speed as the maximum information rate when congestion monitoring is enabled. Also some of the protocols use an interface's configured line speed when calculating a route's cost.

The line speed is not configurable on a Frame Relay dial circuit interface. If the dial circuit is mapped to an ISDN base interface, 64Kbps is used as the line speed.

Minimum Information Rate

The *minimum information rate (IR)* is the minimum data rate for a PVC that the router throttles down to when it is notified of congestion. You set the minimum IR as a percentage of CIR using the **set ir-adjustment** configuration command. It can be dynamically changed using the **set ir-adjustment** console command. If you configure CIR equal to 0, the minimum IR is 64 bps.

Maximum Information Rate

The *maximum information rate* is the maximum data rate at which the router transmits for a PVC. If the CIR monitoring feature is enabled and CIR and Bc are non-zero, the maximum information rate is calculated using CIR, Bc, and Be as follows:

```
( Bc + Be ) * CIR / Bc
```

If the CIR monitoring feature is enabled and CIR and Bc are configured equal to 0, the maximum information rate is equal to the excess burst size (Be).

If the CIR monitoring feature is not enabled the maximum information rate is equal to the line speed.

Variable Information Rate

The *variable information rate* (VIR) ranges from the configured minimum IR to the calculated maximum IR when the CIR monitoring or congestion monitoring features are enabled. The VIR is gradually decreased down to the minimum information rate when the router is notified of congestion on a circuit and is gradually increased to the maximum information rate when the router stops receiving congestion notifications. Using the **set ir-adjustment** configuration command, you configure the percentage of the information rate by which the VIR should decrease when the router is notified of congestion. You also use this command to configure the percentage of the information rate by which the VIR should be gradually increased when the congestion ends.

To avoid impulse loading of the network, the router initially sets the VIR to CIR when the PVC becomes active. If you configure 0 for CIR, VIR is initially set to excess burst (Be) times the MIR adjustment percentage. For example, if Be is set to 64000 and the MIR adjustment percentage is set to 25%, then the initial VIR would be equal to 16000 bps.

The VIR can actually exceed the maximum value in one case. If the length of a frame in bits is greater than the maximum IR, Frame Relay transmits the frame anyway.

Circuit Congestion

Circuit congestion occurs for one of the following reasons:

- The sender is transmitting faster than the allowable throughput
- The receiver is too slow when processing the frames
- An intermediate backbone link is congested, resulting in the sender transmitting faster than the resulting throughput.

When circuit congestion happens, the network must drop packets and/or shut down.

In response to circuit congestion, the router implements a *throttle down*, which is a stepwise slowing of packet transmission to the configured minimum IR. Throttle down occurs during the following conditions:

- Circuit congestion is occurring.
- The router is the sender of frames.
- CIR monitoring or congestion monitoring is enabled.

This section discusses monitoring of Frame Relay data rates and circuit congestion.

CIR Monitoring

CIR monitoring is an optional Frame Relay feature that you can set for each interface to prevent the router from creating congestion conditions in the FR network. CIR monitoring allows the VIR for a PVC to range between the configured minimum and maximum IR.

CIR monitoring is configured with the **enable cir-monitor** configuration command and is disabled by default. CIR monitoring, when enabled, overrides congestion monitoring. You can also dynamically enable and disable CIR monitoring using the **enable cir-monitor** and **disable cir-monitor** console commands.

Congestion Monitoring

Congestion monitoring is an optional feature, set per interface, that allows the VIR of PVCs to vary in response to network congestion. The VIR assumes values between the minimum IR and a maximum IR of the line speed. Congestion monitoring is enabled by default. It can be disabled with the **disable congestion-monitor** configuration command and re-enabled with the **enable congestion-monitor** command. You can also dynamically enable and disable congestion monitoring using the **enable congestion-monitor** and **disable congestion-monitor** console commands.

CIR monitoring, if enabled, overrides congestion monitoring. If both CIR monitoring and congestion monitoring are disabled, the VIR for each PVC on the interface is set to the line speed and does not decrease in response to network congestion.

Congestion Notification and Avoidance

When congestion occurs, the FR backbone network is responsible for notifying the sender and receiver by sending out a FECN or a BECN signal. FECN and BECN are bits that are set in the frame to notify the receiver and sender, respectively, that congestion is occurring.

The example in Figure 26-5 on page 26-13 shows a congestion condition at switch B when frames are sent from router X to router Y. The FR backbone network notifies router X that frames it sends are encountering congestion by setting the BECN bit in frames sent to router X. The FR backbone network also notifies router Y that frames it receives encountered congestion by setting the FECN bit.

When the router receives a frame containing BECN, it is the router's responsibility to throttle down the PVC's VIR (variable information rate) if either CIR monitoring or congestion monitoring is enabled. The router does this gradually as it receives consecutive frames with BECN until either the minimum IR is reached or a frame without BECN arrives. As the router receives consecutive frames without BECN, the VIR gradually rises to the maximum IR.
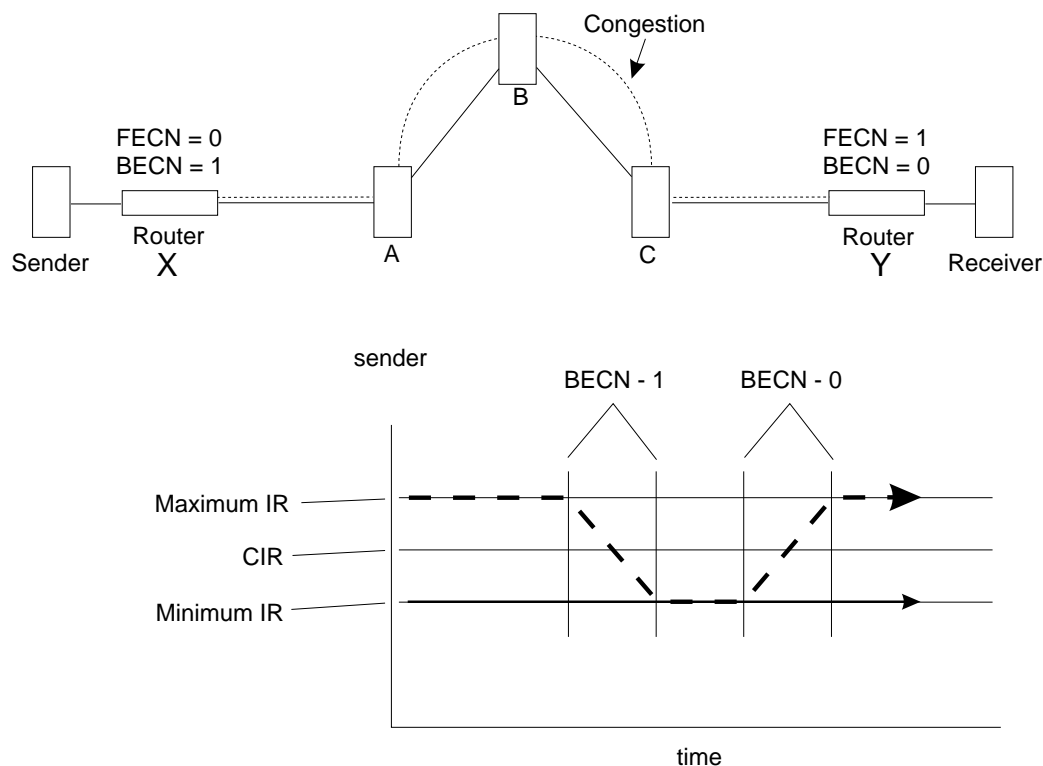
Figure 26-5. Congestion Notification and Throttle Down

Note: If multiple DLCIs are configured between two end-stations when congestion occurs, it is possible that a second DLCI may be used to transmit data at a higher throughput until the congestion condition on the first DLCI is corrected.

Bandwidth Reservation over Frame Relay

For information on bandwidth reservation over Frame Relay, refer to Chapter 10, "Using and Configuring Bandwidth Reservation (BRS) and Priority Queuing" on page 10-1 through Chapter 11, "Monitoring Bandwidth Reservation" on page 11-1.

Displaying the Frame Relay Configuration Prompt

To access the Frame Relay configuration environment:

1. At the OPCON prompt (∗), type **talk 6**.

2. At the configuration prompt (Config>), enter the **list devices** command to see a list of interfaces configured on the router.

3. Enter the **network** command to display the Frame Relay configuration prompt. The network number is the number of the Frame Relay interface.

   ```
   Config>network
   What is the network number [0] 2
   Frame Relay user configuration
   FR Config>
   ```

4. At the Frame Relay interface configuration prompt (FR Config>), use the commands discussed in this chapter to configure Frame Relay parameters.

Frame Relay Basic Configuration Procedure

This section outlines the minimum configuration steps that you are required to perform to get the Frame Relay protocol up and running. If you desire any further configuration information and explanation, refer to the configuration commands described in this chapter.

Note: You must restart the router for new configuration changes to take effect.

- **Select FR management.** The FR Local Management Interface (LMI) protocol defaults to ANSI. You have the option of connecting to a network using the Interim LMI (REV1), ANSI T1.617 Annex D management, or ITU-T/CCITT Q.933 Annex A management. Use the **enable** and **set** commands to enable and set the required management.

- **Add a PVC.** Add any required PVCs that are needed if FR management is disabled or orphan circuits are disabled. If you want to bridge over a FR PVC, or if you want to run APPN over a FR PVC, you also must configure that PVC. Use the **add permanent-virtual-circuit** command.

- **Configure FR destination addresses.** If you are running a protocol such as IP or IPX over the FR interface, and are interconnecting with devices not supporting the Address Resolution Protocol (ARP) or Inverse ARP on FR, use the **add protocol-address** command to add the static protocol and address mapping.

- **Configure Bandwidth Reservation over Frame Relay.** In addition to the basic Frame Relay configuration, which must be done, you can also configure Bandwidth Reservation (an optional feature) over Frame Relay. For information on configuring Bandwidth Reservation, refer to Chapter 10, "Using and Configuring Bandwidth Reservation (BRS) and Priority Queuing" .

Enabling Frame Relay Management

There are three management options under Frame Relay:

- Interim Local Management Interface Revision 1
- ANSI T1.617 Annex D management
- ITU-T/CCITT Q.933 Annex A management.

Frame Relay defaults to ANSI enabled. If you want to change management types, or if you want to re-enable ANSI management, use the following procedure. Enabling management over Frame Relay is a two-step process:

1. Enter the **enable lmi** command at the `FR Config>` prompt to enable management activity.

2. Enter the **set lmi-type** command to select the type of management for the interface.

   See Table 26-2 on page 26-15 for details of the management types available using the **set** command.

   An example of how to set these management types is shown after the table. Also, refer to the **enable** and **set** command sections in this chapter for more information.

Table 26-2. Frame Relay Management Options

Command	Options	Description
set	lmi-type rev1	Conforms to LMI Revision 1 (Stratacom's Frame Relay Interface Specification)
set	lmi-type ansi	Conforms to ANSI T1.617 ISDN-DSS1-Signalling Specification for Frame Relay Bearer Service (known as Annex D)
set	lmi-type ccitt	Conforms to Annex A of ITU-T/CCITT Recommendation Q.933 - DSS1 Signalling Specification for Frame Mode Basic Call Control.

Example: `enable lmi`

   `set lmi-type ansi`

Frame Relay Configuration Commands

This section summarizes and then explains the Frame Relay configuration commands. Enter all commands at the `Frame Relay>` prompt.

You must restart the router for new configuration changes to take effect.

Table 26-3. Frame Relay Configuration Commands Summary

Command	Function
? (Help)	Lists the configuration commands or lists any parameters associated with the commands.
Add	Adds PVCs, Required PVC groups, and destination protocol addresses to the Frame Relay interface.
Change	Modifies a PVC or Required PVC group previously defined by the **add** command.
Disable	Disables any enabled Frame Relay features.
Enable	Enables Frame Relay features such as circuit monitoring, management options, multicast, protocol-broadcast, and orphans.
List	Displays the current configuration of the LMI, PVCs, Required PVC groups, HDLC information, and protocol addresses.
LLC	Configures LLC parameters on the Frame Relay interface. These LLC parameters are required when running APPN over the Frame Relay interface.
Remove	Deletes any previously added PVCs, Required PVC groups (if empty), or protocol addresses.
Set	Configures the Frame Relay management options and parameters (N1-parameter, N2-parameter, N3-parameter, P1 parameter, and T1-parameter). Configures the physical-layer parameters for FR serial interfaces. Sets the maximum frame size.
Exit	Exits the Frame Relay configuration and returns to the `Config>` prompt.

Note: In this section, the terms *circuit number* and *PVC* are synonymous with the term DLCI (Data Link Circuit Identifier).

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
Add
Change
Disable
Enable
List
Remove
Set
Exit
```

Example: **enable ?**

```
cir-monitor
congestion-monitor
DN-length-field
lmi
lower-dtr
multicast-emulation
no-pvc
orphan-circuits
protocol-broadcast
```

Add

Use the **add** command to add a PVC, Required PVC group, or destination protocol address supported by the Frame Relay interface.

Syntax: <u>a</u>dd   <u>pe</u>rmanent-virtual-circuit . . .

                   <u>pr</u>otocol-address . . .

                   <u>pv</u>c-group . . .

permanent-virtual-circuit

Adds a PVC to the Frame Relay interface beyond the reserved range 0 through 15. The maximum number of PVCs that can be added is approximately 992, but the actual number of PVCs that the interface can support depends on the throughput required for each PVC, the line speed, the type of protocols running on the interface, and the number of local management interface PVC information elements that can fit in the maximum frame size.

Example: **add permanent-virtual-circuit**

```
Circuit Number [16]?
Committed Information Rate (CIR) in bps [64000]?
Committed Burst Size (Bc) in bits [64000]?
Excess Burst Size (Be) in bits [0]?
Assign Circuit name []?
Is circuit required for interface operation [N]?
Does the circuit belong to a required PVC group [N]?
What is the group name []?
```

Circuit Number   Indicates the circuit number in the range 16 to 1007.

Committed Information Rate

                        Indicates the committed information rate (CIR). The CIR can be either 0 or a value in the range 300 bps to 2048000 bps. For more information, see "Committed

Information Rate (CIR)" on page 26-9. The default is 64000 bps.

Committed Burst Size   The maximum amount of data in bits that the network agrees to deliver during a measurement interval equal to committed burst (Bc) size / CIR seconds. The range is 300 to 2048000 bits. The default value is 64000 bits.

> **Note:** If CIR is configured as 0 then the committed burst size is set to 0 and you are not prompted for a value. For additional information, see "Committed Burst (Bc) Size" on page 26-9.

Excess Burst Size   The maximum amount of uncommitted data in bits in excess of committed burst size that the network attempts to deliver during a measurement interval equal to (Committed Burst Size/CIR) seconds. Range is 0 to 2048000 bits. Default is 0. For additional information, see "Excess Burst (Be) Size" on page 26-10.

Assign Circuit Name   Indicates the ASCII string that is assigned to describe the circuit. The default is unassigned.

Is the circuit required for operation

   Specify Y or N to indicate whether the circuit is required for interface operation.

Does the circuit belong to a required PVC group

   This prompt is only displayed for circuits that are required.

   Specify Y or N to indicate whether the circuit should belong to a required PVC group.

What is the group name

   Specifies the name of the required PVC group when the PVC is defined as belonging to a required group. Enter a question mark (?) for a list of currently defined groups.

protocol-address

This command adds statically configured destination protocol (protocol-name) addresses to the Frame Relay interface. Statically configured destination protocol addresses are useful if neither Inverse ARP nor ARP is an option, or for other reasons such as security. Adding protocol name and address mappings (static ARP) is less efficient than Inverse ARP or ARP.

- Inverse ARP is the preferred, efficient method because of dynamic address mapping with no broadcasts.

- ARP is recommended if Inverse ARP is not an option. It is less efficient than Inverse ARP because it uses address broadcast and mappings are re-learned at regular intervals.

This parameter prompts you for different information depending on the type of protocol that you are adding.

Example: `add protocol-address`

```
Protocol name or number [0]?
```

IP protocol:

```
IP Address [0.0.0.0]?
Circuit Number [16]?
```

IPX protocol:

```
Host Number (in hex)[]?
Circuit Number [16]?
```

AppleTalk Phase 2 protocol:

```
Network Number (1-65279) []?
Node Number (1-253) []?
Circuit Number [16]?
```

DN protocol:

```
Node address [0.0]?
Circuit Number [16]?
```

Protocol name or number

Defines the name or number of the protocol that you are adding. If you should specify an unsupported protocol, the system will prompt you with the error message:

```
   Unknown protocol name, try again
```

For example, you may have erroneously specified one of the following:

```
Prot#  Name
0      IP
4      DN
7      IPX
22     AP2
```

To see a list of supported protocol types, type ? at the `Protocol name or number [IP]?` prompt.

IP Address   Defines the 32-bit Internet address in dotted-decimal notation of the remote IP host.

Host Number   Defines the 48-bit IPX node address of the remote IPX host.

Network Number   Defines the AppleTalk Phase 2 network number of the remote AppleTalk host.

Node Number   Defines the node number of the interface attached to the remote AppleTalk host.

Node address   Defines the DECnet node address of the remote DECnet host. Configure the node address in the format *x.y*, where *x* is a 6-bit area address and *y* is a 10-bit node number.

Circuit Number   Defines the PVC in the range 16 to 1007 that this protocol is to run over.

pvc-group

Adds a Required PVC group name.

Example: **add pvc-group**

```
PVC group name []? group1
```

Change

Use the **change permanent-virtual-circuit** command to change any previous PVCs that were added with the **add permanent-virtual-circuit** command.

Syntax: change permanent-virtual-circuit . . .

Example: **change permanent-virtual-circuit**

```
Circuit Number [16]?
Committed Information Rate in bps [64000]?
Committed Burst Size (Bc) in bits [64000]?
Excess Burst Size (Be) in bits [0]?
Assign Circuit Name: []?
Is the circuit required for interface operation [N]?
Does the circuit belong to a required group [N]?
What is the group name []?
```

Circuit Number   Indicates the circuit number in the range 16 to 1007.

Committed Information Rate
                            Indicates the committed information rate (CIR). The CIR can be either 0 or a value in the range 300 bps to 2048000 bps. The default is 64000 bps.

Committed Burst Size   The maximum amount of data in bits that the network agrees to deliver during a measurement interval equal to (Committed Burst Size/CIR) seconds. If the CIR is configured as 0, the committed burst size is also set to 0. Otherwise, the range of valid values is 300 to 2048000 bits. The default is 64000 bits.

Excess Burst Size   The maximum amount of uncommitted data in bits in excess of Committed Burst Size that the network attempts to deliver during a measurement interval equal to (Committed Burst Size/CIR) seconds. Range is 0 to 2048000 bits. Default is 0.

Assign circuit Name   Indicates the ASCII character string designation for the circuit that you want to change.

Is the circuit required for operation
                            Specify Y or N to indicate whether the circuit is required for interface operation.

Does the circuit belong to a required PVC group
                            This prompt is only displayed for circuits that are required.

                            Specify Y or N to indicate whether the circuit should belong to a required PVC group.

What is the group name
                            Specifies the name of the required PVC group when the PVC is defined as belonging to a required group. Enter a question mark (?) for a list of currently defined groups.

Disable

Use the **disable** command to disable those features previously enabled using the **enable** command.

Syntax: <u>d</u>isable   <u>ci</u>r-monitor

                    <u>co</u>ngestion-monitor

                    <u>d</u>n-length-field

                    <u>l</u>mi

                    lower-dtr

                    <u>m</u>ulticast-emulation

                    <u>no</u>-pvc

                    <u>o</u>rphan-circuits

                    <u>p</u>rotocol-broadcast

cir-monitor

     Disabling this feature allows the circuit's information rate to exceed the maximum information rate that is calculated using the parameters configured with the **add permanent-virtual-circuit** command. The default setting for this feature is disabled. See "Circuit Congestion" on page 26-11 for more information.

     Example: `disable cir-monitor`

congestion-monitor

     Disables the congestion monitoring feature. Disabling this feature prevents a circuit's information rate from varying in response to congestion between the minimum information rate and the line speed. See "Circuit Congestion" on page 26-11 for more information. The default setting for this feature is enabled.

     Example: `disable congestion-monitor`

dn-length-field

     Prevents interoperation with implementations of DECnet Phase IV over Frame Relay that require a length field to precede DECnet packets in Frame Relay frames, but allows interoperation with DECnet Phase IV Frame Relay software that does not use a length field before the DECnet packet. Disabling dn-length-field causes Frame Relay not to insert a length field into transmitted frames containing DECnet packets and not to attempt to remove the length field from received frames containing DECnet packets.

     Note: This option is presented as a configuration option only

     Example: `disable dn-length-field`

lmi

     Disables management activity. All circuits that were statically added are marked as present and active from the network perspective. The default for this feature is enabled, and the default lmi type is ANSI T1.617 Annex D management.

     Note: Disabling this parameter allows for normal operation or end-to-end Frame Relay testing in the absence of a real network or management interface. With end-to-end Frame Relay testing, it is necessary to add like PVCs (the same PVC number, such as 16 and 16) on both ends of the link.

     Example: `disable lmi`

lower-dtr

> This parameter determines how the data terminal ready (DTR) signal is
> handled for leased serial-line interfaces on the router. It is not supported on
> Frame Relay dial circuit interfaces. See the **enable lower-dtr** command for a
> more complete description of the lower-dtr parameter.
>
> The following cable types are supported:
>
>> EIA 232 (RS-232)
>> V.35
>> V.36
>
> The default setting is **disable lower-dtr**.
>
> **Example:**　`disable lower-dtr`

multicast-emulation

> Disables multicast emulation on each active PVC. The default setting for this
> feature is enabled. If you disable this feature, you are required to add protocol
> static address maps.
>
> Some protocols, such as IPX RIP, will not function on the Frame Relay
> interface if multicast-emulation is disabled. The protocol-broadcast feature also
> requires multicast-emulation in order to function properly. For more
> information, see "Multicast Emulation and Protocol Broadcast" on page 26-7.
>
> **Example:**　`disable multicast-emulation`

no-pvc

> Controls whether the interface is considered active or inactive. If no-pvc is
> disabled, the presence of active PVCs on the interface does not affect whether
> the Frame Relay interface is considered active or inactive.

orphan-circuits

> Prohibits the use of all non-configured orphan circuits at the interface. The
> default setting for orphan circuits is enabled. Disabling orphan circuits adds a
> measure of security to your network by preventing unauthorized entry from a
> non-configured circuit. However, if you disable orphan circuits, you are
> required to add PVCs.
>
> **Example:**　`disable orphan-circuits`

protocol-broadcast

> Prohibits protocols such as IP RIP from functioning over the Frame Relay
> interface. For more information, see "Multicast Emulation and Protocol
> Broadcast" on page 26-7. The default setting for this feature is enabled.
>
> **Example:**　`disable protocol-broadcast`

Enable

Use the **enable** command to enable Frame Relay features.

Syntax:　en‍able　　c‍ir-monitor
　　　　　　　　　　co‍ngestion-monitor
　　　　　　　　　　d‍n-length-field
　　　　　　　　　　l‍mi
　　　　　　　　　　lower-dtr
　　　　　　　　　　m‍ulticast-emulation
　　　　　　　　　　o‍rphan-circuits
　　　　　　　　　　n‍o-pvc

protocol-broadcast

cir-monitor

Enables the circuit monitoring feature. The circuit monitoring feature ensures that the circuit's information rate varies between the minimum information rate and the maximum information rate, calculated using the parameters configured with the **add permanent-virtual-circuit** command or the **change permanent-virtual-circuit** command

Note: The circuit monitoring feature overrides the congestion monitoring feature if there is a conflict when both are enabled. The default setting for this feature is disabled.

For additional information on CIR monitoring, see "CIR Monitoring" on page 26-12.

Example: `enable cir-monitor`

congestion-monitor

Enables the congestion monitoring feature. This feature allows a circuit's information rate to vary in response to congestion between the minimum information rate and the line speed.

Note: The circuit monitoring feature overrides the congestion monitoring feature if there is a conflict when both are enabled. The default setting for this feature is enabled.

For additional information on congestion monitoring, see "Congestion Monitoring" on page 26-12.

Example: `enable congestion-monitor`

dn-length-field

Supports interoperation with implementations of DECnet Phase IV over Frame Relay that require a length field to precede DECnet packets in Frame Relay frames. Enabling dn-length-field causes Frame Relay to insert a length field into transmitted frames containing DECnet packets and to remove the length field from received frames containing DECnet packets. This option is disabled by default. By default, Frame Relay will neither insert nor attempt to remove the length field.

Note: This option is presented as a configuration option only when the router software contains the DECnet Phase IV protocol.

Example: `enable dn-length-field`

lmi

Enables management activity.

After issuing the **enable lmi** command, use the **set lmi-type** command to select the management mode for your Frame Relay interface. See "Enabling Frame Relay Management" on page 26-14. The system defaults to ANSI T1.617 Annex D management.

Use the **enable lmi** command to resume LMI management if you have previously disabled Frame Relay management.

Example: `enable lmi`

lower-dtr

This parameter determines how the data terminal ready (DTR) signal is handled for leased serial-line interfaces that are disabled. It is not supported

on Frame Relay dial circuit interfaces. If this parameter is set to "disabled" (the default), the DTR signal will remain raised when the interface is disabled.

When lower-dtr is enabled, DTR will be lowered when the interface is disabled. This behavior may be desirable in situations where the interface has been configured as an alternate link for WAN Reroute and the interface is connected to a dial-out modem which maintains its dial connection based on the state of the DTR signal.

If this feature is enabled and the interface is disabled, the DTR signal is low and the modem keeps the dial connection down. When the interface is enabled, due to a WAN Reroute backup scenario, DTR is raised and the modem dials a stored number to the backup site. When the primary interface is restored, the alternate interface is disabled, DTR is lowered, and the modem hangs up the dial connection.

The following cable types are supported:

> EIA 232 (RS-232)
> V.35
> V.36

The default setting is **disable lower-dtr**.

Example: `enable lower-dtr`

multicast-emulation
> Enables multicast emulation. This allows a multicast/broadcast frame to be transmitted on each active PVC. Protocols such as ARP, IPX RIP, and IP RIP require multicast emulation to be enabled to function correctly over a Frame Relay interface. For more information, see "Multicast Emulation and Protocol Broadcast" on page 26-7. The default for this parameter is enabled.

> **Example:** `enable multicast-emulation`

no-pvc
> Controls whether the interface is considered active or inactive. When this feature is enabled, the Frame Relay interface becomes inactive when there are no active PVCs on the interface. If at least one PVC is active, the Frame Relay interface becomes active when a successful LMI exchange occurs between the router and the FR switch.

orphan-circuits
> Enables the use of all non-configured orphan circuits. The default for this feature is enabled. The CIR defaults to 64000 bps, the Committed Burst (Bc) Size to 64000 bps, and the Excess Burst (Be) Size to 0 for each orphan circuit.

> **Example:** `enable orphan-circuits`

protocol-broadcast
> Allows protocols such as IP RIP to function correctly over the Frame Relay interface. The multicast emulation feature must be enabled for the protocol-broadcast feature to function correctly. The default setting for this feature is enabled.

> **Example:** `enable protocol-broadcast`

List

Use the **list** command to display currently configured management and PVC information.

Syntax: l̲ist   all
                        h̲dlc
                        l̲mi
                        p̲e̲rmanent-virtual-circuits
                        p̲r̲otocol-address
                        p̲v̲c-groups

all

Displays the Frame Relay configuration.

See **list hdlc** and **list lmi** for descriptions of the parameters.

Example: `list all`

```
                        Frame Relay HDLC Configuration

Encoding     =  NRZI              IDLE                  = Flag
Clocking     =  External
Cable type   =  V.35 DTE
Line speed (bps)      = 64000    Interface MTU in bytes  = 2048
Transmit delay        = 0
Lower DTR             = Disabled

                        Frame Relay Configuration


LMI enabled  =   Yes  LMI DLCI      = 1023
LMI type     =  REV1  LMI Orphans OK =  Yes

Protocol broadcast    =   Yes  Congestion monitoring       =  Yes
Emulate multicast     =   Yes  CIR monitoring              =  No


PVCs P1 allowed       =   64  Interface down if no PVCs    =  No
Timer T1 seconds      =   10  Counter N1 increments        =   6
LMI N2 error threshold =    3  LMI N3 error threshold window =    4
MIR % of CIR          =   25  IR % Increment               =  12
IR % Decrement        =   25  DECnet length field          =  No


Maximum PVCs allowable =       64
Total PVCs configured  =        7
            Circuit              Circuit  Circuit   CIR    Burst  Excess
            Name                 Number   Type     in bps  Size   Burst
-------------------------------- ------- ---------- ------- ------- -------
  cir16                             16   #Permanent  64000   64000     0
  cir244                           244   #Permanent  64000   64000     0
  cir33                             33   #Permanent  64000   64000     0
  cir1005                         1005   #Permanent  64000   64000     0
  cir55                             55    Permanent  64000   64000     0
  cir22                             22   #Permanent  64000   64000     0
  cir66                             66   *Permanent  64000   64000     0

* = circuit is required
# = circuit is required and belongs to a required PVC group

Required PVC group = group1

      Circuit #  16
      Circuit #  244
      Circuit #  22
```

```
Required PVC group = group2

     Circuit #  33
     Circuit #  1005


No address translations configured
```

hdlc

Displays the Frame Relay High-Level Data Link Control (HDLC) configuration.

Example: `list hdlc`

```
                    Frame Relay HDLC Configuration

Encoding     = NRZI              IDLE                = Flag
Clocking     = External
Cable type   = V.35 DTE
Line speed (bps)     = 64000    Interface MTU in bytes  = 2048
Transmit Delay       = 0
Lower DTR            = Disabled
```

Encoding	The transmission encoding scheme for the serial interface. Encoding is NRZ (non-return to zero) or NRZI (non-return to zero inverted).
Idle	The data link idle state: flag or mark.
Clocking	The type of clocking: internal or external.
Cable type	The serial adapter cable type: RS-232, V.35, V.36, or X.21.
Line Speed (bps)	Indicates the physical data rate for the Frame Relay interface.
Interface MTU in bytes	Indicates the maximum transmission unit (amount of user data per frame) that can be transmitted or received over the network at any given time.
Transmit delay	Indicates the number of flag bytes sent between frames.
Lower DTR	Indicates whether the router will drop the DTR signal when a WAN Reroute alternate link is no longer needed. Dropping the DTR signal causes the modem to terminate the leased-line connection for the alternate link. Lower DTR does not appear when the cable type is X.21.

Note: For a FR dial circuit interface, only the Interface MTU is displayed.

lmi

Displays logical management and related configuration information about the Frame Relay interface.

Example: `list lmi`

Configuring Frame Relay Interfaces

```
                              Frame Relay Configuration


          LMI enabled            =  Yes  LMI DLCI                   =    0
          LMI type               = ANSI  LMI Orphans OK             =  Yes


          Protocol broadcast     =  Yes  Congestion monitoring      =  Yes
          Emulate multicast      =  Yes  CIR monitoring             =   No

          PVCs P1 allowed        =   64  Interface down in no PVCs  =   No
          Timer T1 seconds       =   10  Counter N1 increments      =    6
          LMI N2 error threshold =    3  LMI N3 error threshold window =  4
          MIR % of CIR           =   25  IR % Increment             =   25
          IR % Decrement         =   25  DECnet length field        =   No
```

LMI enabled	Indicates whether the management features are enabled on the Frame Relay interface, yes or no.
LMI DLCI	Indicates the management circuit number. This number reflects the LMI type: 0 for ANSI and ITU-T/CCITT and 1023 for REV1.
LMI Type	Indicates the LMI type: REV1, ANSI, or CCITT.
LMI Orphans OK	Indicates if non-configured circuits are available for use, yes or no.
Protocol Broadcast	Indicates whether protocols such as IP RIP can function over the Frame Relay interface, yes or no.
Emulate multicast	Indicates whether the multicast emulation feature is enabled on each active PVC, yes or no.
Congestion Monitoring	Indicates whether the congestion monitoring feature that responds to network congestion is enabled, yes or no.
CIR monitoring	Indicates whether the circuit monitoring feature that enforces the transmission rate is enabled, yes or no.
PVCs P1 allowed	Indicates the number of allowable PVCs for use with this interface.
Timer T1 seconds	Indicates the frequency with which the Frame Relay interface performs a sequence number exchange with the Frame Relay switch LMI entity.
Counter N1 increments	Indicates the number of T1 timer intervals which must expire before a complete PVC LMI status enquiry is made.
LMI N2 error threshold	Indicates the number of management event errors occurring within the N3 window that will cause a reset of the Frame Relay interface.
LMI N3 error threshold window	Indicates the number of monitored management events used to measure the N2 error threshold.
MIR % of CIR	Minimum IR, expressed as a percentage of CIR.
IR % Increment	Percentage by which the router increments the IR each time it receives a frame without BECN until it reaches the maximum IR.

IR % Decrement Percentage by which the router decrements the IR each time it receives a frame that contains BECN until it reaches the minimum IR.

permanent-virtual-circuits
    Displays all the configured PVCs on the Frame Relay interface.

Example: `list permanent-virtual-circuit`

```
FR Config>li perm

 Maximum PVCs allowable =      64
 Total PVCs configured  =       7

            Circuit            Circuit  Circuit   CIR     Burst Excess
            Name               Number    Type    in bps   Size  Burst
-------------------------------- ------- ---------- ------- ------- -------
cir16                               16   #Permanent  64000   64000      0
cir244                             244   #Permanent  64000   64000      0
cir33                               33   #Permanent  64000   64000      0
cir1005                           1005   #Permanent  64000   64000      0
cir55                               55   #Permanent  64000   64000      0
cir22                               22    Permanent  64000   64000      0
cir66                               66   *Permanent  64000   64000      0

* = circuit is required
# = circuit is required and belongs to a Required PVC group
```

Maximum PVCs allowable
    Indicates the number of PVCs that can exist for this interface. This number includes any PVCs that you added with the **add permanent-virtual-circuit** command and dynamically learned through the management interface.

Total PVCs configured Indicates the total number of currently configured PVCs for this interface.

Circuit Name Indicates the ASCII designation of the configured PVC.

Circuit Number Indicates the number of a currently configured PVC.

Circuit Type Indicates the type of virtual circuit currently configured. This release of Frame Relay only supports permanent virtual circuits.

Committed Information Rate
    Indicates the information rate at which the network agrees to transfer data under normal conditions.

Committed Burst Size The maximum amount of data in bits that the network agrees to deliver during a measurement interval equal to (Committed Burst Size/CIR) seconds.

Excess Burst Size The maximum amount of uncommitted data in bits in excess of Committed Burst Size that the network attempts to deliver during a measurement interval equal to (Committed Burst Size/CIR) seconds.

pvc-groups
> Displays all the Required PVC groups on the Frame Relay interface.

> **Example:** `list pvc-groups`

```
Required PVC group = group1

    Circuit # 16
```

protocol-addresses
> Displays all the statically configured protocol addresses of circuit mappings at the Frame Relay interface.

> **Example:** `list protocol-addresses`

```
         Frame Relay Protocol Address Translations

  Protocol Type          Protocol Address    Circuit Number
  -------------          ----------------    --------------
       IP                  125.2.29.4             21
       IPX                 000000004503           16
```

> *Protocol Type*   Displays the name of the protocol running over the interface.

> *Protocol Address*   Displays the protocol address of the device at the other end of the circuit.

> *Circuit Number*   Displays the PVC that is handling the protocol.

LLC

Use the **LLC** command to access the LLC configuration environment. See "LLC Configuration Commands" on page 19-1 for an explanation of each of these commands.

Note: The **LLC** command is supported only if APPN is in the software load.

Syntax: <u>llc</u>

Example: `llc`
> `LLC config>`

Remove

Use the **remove** command to delete any PVC, Required PVC group, or protocol-address previously added using the **add** command.

Syntax: <u>r</u>emove   <u>pe</u>rmanent-virtual-circuit . . .
           <u>pr</u>otocol-address
           <u>pv</u>c-group

permanent-virtual-circuit *pvc#*

Deletes any configured PVC in the range 16 to 1007.

> **Example:** `remove permanent-virtual-circuit 20`

protocol-address
> Deletes any configured protocol addresses (static ARP entries). This parameter prompts you for different information depending on the type of protocol that you are adding.

Example: **remove protocol-address**

```
Protocol name or number [IP]?
```

pvc-group

Deletes any configured PVC group by name. The group is removed only if it has no member circuits.

> **Example:**   **remove pvc-group**
>
> ```
> PVC group name [IP]?
> ```

IP protocol:

```
IP Address [0.0.0.0]?
Circuit Number [16]?
```

IPX protocol:

```
Host Number (in hex)[]?
Circuit Number [16]?
```

AppleTalk Phase 2 protocol:

```
Network Number (1-65279) []?
Node Number (1-253) []?
Circuit Number [16]?
```

DN protocol:

```
Node address [0.0]?
Circuit Number [16]?
```

Protocol name or number

Defines the name or number of the protocol that you are deleting. If you try to delete an unsupported protocol the system will display the error message:

```
Unknown protocol name, try again
```

To see a list of supported protocols, type ? at the `Protocol name or number [IP]?` prompt.

IP Address   Defines the 32-bit internet address of the remote IP host in dotted-decimal notation.

Host Number   Defines the 48-bit node address of the remote IPX host.

Network Number   Defines the AppleTalk Phase 2 network number.

Node Number   Defines the node number of the interface attached to the remote AppleTalk host.

Node address   Defines the DECnet node address of the remote DECnet host. Configure the node address in the format *x,y*, where *x* is a 6-bit area address and *y* is a 10-bit node number.

Circuit Number   Defines the PVC in the range 16 to 1007 that the protocol runs over.

Set

Use the **set** command to configure the interface to run the Frame Relay protocol.

Set Command Considerations

Two parameters, the n2-parameter and the n3-parameter, require further explanation before you configure them. The n2-parameter sets the error threshold for management events, and the n3-parameter sets the number of events that are monitored in the event window. If the number of management errors in the event window equals n2, the Frame Relay interface resets. For example:

set n3-parameter 4
set n2-parameter 3

You now have a window size of 4 (n3 = 4) and an error threshold of 3 (n2 = 3). That means the system is monitoring 4 management events and checking to determine if any of those are in error. If the number of events in error equals 3 (the n2 parameter), the Frame Relay interface is reset and the status of the network is considered *network down*.

For the status of the network to be considered *network up*, the number of events in error within the window must be less than n2 prior to any change in status.

Syntax: <u>set</u>   <u>c</u>able*
                    <u>c</u>locking*
                    <u>e</u>ncoding*
                    <u>f</u>rame-size
                    <u>i</u>dle . . .*
                    <u>ir</u>-adjustment . . .
                    <u>l</u>ine-speed*
                    <u>lm</u>i-type
                    <u>n1</u>-parameter
                    <u>n2</u>-parameter
                    <u>n3</u>-parameter
                    <u>p</u>1-parameter
                    <u>t</u>1-parameter
                    <u>t</u>ransmit-delay . . .*

* Note: The commands with an * following them are not available for FR dial circuit interfaces.

cable *physical-interface-link-type data-connection-type*
    Sets the cable type for the network physical link.

    A DTE cable is used when you are attaching the router to some type of DCE device (for example, a modem or a DSU/CSU). A DCE cable is used when the router is acting as the DCE and providing the clocking for direct attachment. The available options are:

Physical Interface Link Type	Data Connection Type
EIA 232 (RS-232)	DTE, DCE
V35	DTE, DCE
V36	DTE, DCE
X21	DTE, DCE

> **Example:** `set cable rs-232 dte`

clocking *external* or *internal*

> To connect to a modem or DSU, configure clocking as external. To connect
> directly to another DTE device, use a DCE cable and set the clocking to
> internal. For internal clocking, you must enter the **set line-speed** command to
> configure a clock speed using the ranges shown in Table 26-4 on page 26-32.
>
> **Example:** `set clocking internal`

encoding *NRZ* or *NRZI*

> Sets the HDLC transmission encoding scheme as NRZ (non-return to zero) or
> NRZI (non-return to zero inverted). Most configurations use NRZ, which is the
> default.
>
> **Example:** `set encoding nrz`

frame-size *#*

> Sets the size of the network layer portion of frames transmitted and received
> on the data link. This size includes the two bytes containing the DLCI and
> user data. See Figure 26-4 on page 26-5 for more information. Values are 5
> to 8190. The default is 2048.
>
> **Example:** `set frame-size 2000`

idle *flag* or *mark*

> Sets the transmit idle state for HDLC framing. The default value is **flag**, which
> provides continuous flags (7E hex) between frames. The mark option puts the
> line in a marking state (OFF, 1) between frames.
>
> **Example:** `set idle flag`

ir-adjustment *increment-% decrement-% minimum-IR*

> Sets the minimum information rate (IR) and the percentages for incrementing
> and decrementing the IR in response to network congestion.
>
> The minimum IR, expressed as a percentage of CIR, is the lower limit of the
> information rate. The minimum percentage is 1 and the maximum percentage
> is 100. The default is 25.
>
> When network congestion clears, the information rate is gradually incremented
> by the IR adjustment increment percentage until the maximum information rate
> is reached. The minimum percentage is 1 and the maximum percentage is
> 100. The default is 12.
>
> When network congestion occurs, the information rate is decremented by the
> IR adjustment decrement percentage each time a frame containing BECN is
> received until the minimum information rate is reached. The minimum
> percentage is 1, and the maximum percentage is 100. The default is 25.
>
> **Example: `set ir-adjustment`**
>
> ```
> IR adjustment % increment [1₂]?
> IR adjustment % decrement [25]?
> Minimum IR as % of CIR [25]?
> ```

line-speed *rate*

> For internal clocking, this command specifies the speed of the transmit and
> receive clock lines. Refer to Table 26-4 on page 26-32 for the supported
> speeds.

Table 26-4. Line Speeds When Internal Clocking is Used for 2216 Interfaces	
Adapter Type	**Speed Range**
8-port EIA 232	9600 to 64 000 bps
6-port V.35/V.36	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps
8-port X.21	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps

For external clocking, this command does not affect the hardware (in other words, the actual speed of the line) but it sets the speed some protocols, such as IPX, use to determine routing cost parameters. Congestion monitoring also uses the configured line speed to determine the maximum information rate. Therefore, it is recommended that you set the speed to match the actual line speed. Use Table 26-5 to determine the line speeds supported for the serial adapters when external clocking is used.

Table 26-5. Line Speeds When External Clocking is Used for 2216 Interfaces	
Adapter Type	**Speed Range**
8-port EIA 232	2400 to 64 000 bps
6-port V.35/V.36	2400 to 2 048 000 bps
8-port X.21	2400 to 2 048 000 bps

> **Example:** `set line-speed 64000`

lmi-type *rev1* or *ansi* or *ccitt*

> Sets the management type for the interface. See "Enabling Frame Relay Management" on page 26-14 for details on setting Frame Relay management. The default is type *ansi* enabled.

Table 26-6. Frame Relay Management Options		
Command	**Management Type**	**Description**
set	lmi-type rev1	Conforms to LMI Revision 1, (Stratacom's Frame Relay Interface Specification)
set	lmi-type ansi	Conforms to ANSI T1.617 ISDN-DSS1-Signalling Specification for Frame Relay Bearer Service (known as Annex D)
set	lmi-type ccitt	Conforms to Annex A of ITU-T/CCITT Recommendation Q.933 - DSS1 Signalling Specification for Frame Mode Basic Call Control.

> **Example:** `set lmi-type rev1`
>   *or*
> `set lmi-type ansi`
>   *or*
> `set lmi-type ccitt`

n1-parameter *count*

> Configures the number of T1 timer intervals which must expire before a complete PVC status enquiry is made. *Count* is the interval in the range 1 to 255. The default is 6.

Example: `set n1-parameter`

> `Parameter N1 [6]?`

n2-parameter *max#*

Configures the number of errors that can occur in the management event window monitored by the n3-parameter before the Frame Relay interface resets. Max# is a number in the range 1 to 10. The default is 3. This parameter must be less than or equal to the n3-parameter or you will receive an error message.

Example: `set n2-parameter`

> `Parameter N2 [3]?`

n3-parameter *max#*

Configures the number of monitored management events for measuring the n2-parameter. Max# is a number in the range 1 to 10. The default is 4.

Example: `set n3-parameter`

> `Parameter N3 [4]?`

p1-parameter *max#*

Configures the maximum number of PVCs supported by the Frame Relay interface. This includes active, inactive, removed, and congested PVCs. Max# is a number in the range 0 to 992. The default is 64. 0 (zero) implies that the interface supports no PVCs.

Example: `set p1-parameter`

> `Parameter P1 [64]?`

t1-parameter *time*

Configures the interval (in seconds) between sequence number exchanges with Frame Relay management. The management's T2 timer is the allowable interval for an end station to request a sequence number exchange with the manager. The T1 interval must be less than the T2 interval of the network. *Time* is a number in the range 5 to 30. The default is 10.

Example: `set t1-parameter`

> `Parameter T1 [10]?`

transmit-delay *#*

Allows the insertion of a delay between transmitted packets. The purpose of this command is to slow the serial line so that it is compatible with older, slower serial devices at the other end. It can also prevent the loss of serial line hello packets between the lines. *#* is between 0 and 15 extra flags. The default is zero (0). Setting this parameter provides 0 to 15 extra flags between transmit frames. Table 26-7 lists the units and range values for serial interfaces.

Table 26-7. Transmit Delay Units and Range for the 2216 Serial Interface

Unit	Minimum	Maximum
Extra Flags	0	15

Example: `set transmit-delay 15`

Exit

Use the **exit** command to return to the Config> prompt.

Syntax: <u>ex</u>it

Example: `exit`

Chapter 27. Monitoring Frame Relay Interfaces

This chapter describes the Frame Relay console commands and includes the following sections:

- "Displaying the Frame Relay Console Prompt"
- "Frame Relay Console Commands"
- "Frame Relay Interfaces and the GWCON Interface Command" on page 27-9

Note: For information on monitoring bandwidth reservation over Frame Relay, refer to Chapter 11, "Monitoring Bandwidth Reservation" on page 11-1.

Displaying the Frame Relay Console Prompt

To access the Frame Relay console commands and to monitor Frame Relay on your router, perform the following steps:

1. At the OPCON prompt (∗), type **talk 5**.

2. At the GWCON prompt (+), enter the **interface** command to see a list of interfaces configured on the router.

3. Enter the **network** command followed by the network number of the frame relay interface. For example:

```
+ net 2
Frame Relay Console
FR>
```

Frame Relay Console Commands

This section summarizes and then explains the Frame Relay Console commands. Use these commands to gather information from the database. Table 27-1 shows the commands.

Command	Function
? (Help)	Displays all the Frame Relay console commands (clear and list) or any options associated with those commands.
Clear	Clears statistical information on the Frame Relay interface.
Disable	Disables CIR monitoring and congestion monitoring on the Frame Relay interface.
Enable	Enables CIR monitoring and congestion monitoring on the Frame Relay interface.
List	Displays statistics specific to the data-link layer and Frame Relay management.
LLC	Displays the LLC console monitoring prompt.
Set	Sets CIR, Committed Burst Size, and Excess Burst Size for a Frame Relay PVC.
Exit	Exits the Frame Relay console process.

Table 27-1. Frame Relay Console Commands Summary

Note: In this section, the terms *circuit number* and *PVC* are equivalent to the term *data link circuit identifier (DLCI)*.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
CLEAR
DISABLE
ENABLE
LIST
SET
EXIT
```

Clear

Use the **clear** command to remove all statistics on the Frame Relay interface.

Note: Statistics can also be cleared by using the OPCON **clear** command.

Syntax: c̲lear

Example: **clear**

Disable

Use the **disable** command to disable the Frame Relay CIR monitoring and congestion monitoring features.

The **disable** command dynamically changes the router configuration. These changes will be lost when the router is restarted.

Syntax: d̲isable   c̲ir-monitor
                       co̲ngestion-monitor

Example: disable cir-monitor

Enable

Use the **enable** command to enable the Frame Relay CIR monitoring and congestion monitoring features.

The **enable** command dynamically changes the router configuration. These changes will be lost when the router is restarted.

Syntax: e̲nable c̲ir-monitor
                   co̲ngestion-monitor

Example: enable cir-monitor

List

Use the **list** command to display statistics specific to the data-link layer and the Frame Relay interface.

Syntax: l̲ist   a̲ll
                    c̲ircuit . . .
                    l̲mi
                    pe̲rmanent-virtual-circuits
                    pv̲c-groups

all

> Displays circuit, management, and PVC statistics on the Frame Relay interface. The output displayed for this command is a combination of the **list lmi** and **list permanent-virtual-circuit** commands.
>
> **Example:**   `list all`

circuit *pvc#*

> Displays detailed PVC configuration and statistical information for the specified PVC (pvc#).
>
> **Example:**   `list circuit 347`

```
Circuit name = Valencia

Circuit state        =     Active  Circuit is orphan   =    No
Frames transmitted   =          0  Bytes transmitted   =    0
Frames received      =          0  Bytes received      =    0
Total FECNs          =          0  Total BECNs         =    0
Times congested      =          0  Times Inactive      =    0
CIR in bits/second   =      64000  Potential Info Rate =    56000
Committed Burst (BC) =       1200  Excess Burst (Be)   =    54800
Minimum Info Rate    =      16000  Maximum Info Rate   =    64000
Required             =        Yes  PVC group name      =    group1

Current number of xmit frames queued                  =    0
Xmit frames dropped due to queue overflow             =    0
```

Circuit state	Indicates the state of the circuit: inactive, active, or congested. Inactive indicates that the circuit is not available for traffic because either the Frame Relay interface is down or the Frame Relay management entity has not notified the Frame Relay protocol that the circuit is active. Active indicates that data is being transferred. Congested indicates that data flow is being controlled.
Circuit is orphan	Indicates if the circuit is a nonconfigured circuit learned through LMI management.
Frames/Bytes transmitted	Indicates how many frames and bytes this PVC has transmitted.
Frames/Bytes received	Indicates how many frames and bytes this PVC has received.
Total FECNS	Indicates the number of times that this PVC has been notified of inbound or downstream congestion.
Total BECNS	Indicates the number of times that this PVC has been notified of outbound or upstream congestion.
Times congested	Indicates the number of times that this PVC has become congested.
Times inactive	Indicates the number of times that this PVC was inoperable.
CIR in bits/sec	Indicates the information rate of the PVC between the range 300 bps to 2048000 bps. A value of 0 is also supported.

Potential Info Rate   Indicates the current maximum rate in bits per second at which data will be transmitted for the circuit. The actual data rate will depend on the queue depths and priorities associated with the circuit.

If this field has a value of "Line Speed," then the maximum data rate is the actual line speed even if the line speed was not configured or was configured incorrectly for this interface.

Committed Burst (BC)   Maximum amount of data, in bits, that the network commits to deliver during a calculated *time interval* (Tc). (Tc=Bc/CIR.)

Excess Burst (Be)   Maximum amount of uncommitted data the router can transmit on a PVC in excess of the Bc during the time interval (Tc).

Minimum Info Rate   Minimum Information Rate. The minimum data rate for a PVC that the router throttles down to when it is notified of congestion.

Maximum Info Rate   Maximum Information Rate. The maximum data rate at which the router transmits for a PVC.

Required   Yes or No. If yes, the PVC is a Required PVC.

PVC group name   If the PVC is a member of a required PVC group, the name appears here; otherwise, "Unassigned" appears.

Current number of xmit frames queued
    Indicates the number of frames currently queued for this circuit by FR. These frames are waiting for space to become available on the serial device handler transmit queue for this interface.

Xmit frames dropped due to queue overflow
    Indicates the number of frames that could not be transmitted for this PVC due to output queue overflow.

lmi
Displays statistics relevant to the logical management on the Frame Relay interface.

Example: list lmi

```
Management Status:
------------------
  LMI enabled   =   Yes  LMI DLCI       = 1023
  LMI type      = REV1  LMI Orphans OK  =  Yes

  Protocol broadcast   =     Yes  Congestion monitoring =  Yes
  Emulate multicast    =     Yes  CIR monitoring        =   No
  PVCs P1 allowed      =      64  Interface down if no PVCs =  No
  Line speed (bps)     =   64000  Interface MTU in bytes = 2048
  Timer T1 seconds     =      10  Counter N1 increments  =    6
  LMI N2 threshold     =       3  LMI N3 threshold window =   4
  MIR % of CIr         =      25  IR % Increment         =   12
  IR % Decrement       =      25  DECnet length field    =   No

  Current receive sequence  =   0
  Current transmit sequence =   0
  Total status enquires     =   0  Total status responses =   0
  Total sequence requests   =   0  Total responses        =   0
```

```
PVC Status:
-----------
  Total allowed  =      64  Total configured  =      3
  Total active   =       0  Total congested   =      0
  Total left net =       0  Total join net    =      0
```

Management Status:

LMI enabled	Indicates if Frame Relay management is active (yes or no).
LMI DLCI	Indicates the management circuit number. This number is either 0 (ANSI default or ITU-T/CCITT) or 1023 (interim LMI REV1).
LMI type	Indicates the type of frame relay management being used, ANSI, ITU-T/CCITT, or LMI Revision 1.
LMI orphans OK	Indicates if all nonconfigured circuits learned from Frame Relay management are available for use (yes or no).
Protocol broadcast	Indicates if protocols such as IP RIP are able to operate over the Frame Relay interface.
Congestion monitoring	Indicates whether the congestion monitor feature that responds to network congestion is enabled (yes or no).
Emulate multicast	Indicates whether the multicast emulation feature is enabled on each active PVC (yes or no).
CIR monitoring	Indicates whether the circuit monitoring feature that enforces the transmission rate is enabled (yes or no).
PVCs P1 allowed	Indicates the number of allowable PVCs for use with this interface. This number is the maximum number of active, congested, inactive, and removed PVCs that can be supported on the interface.
Interface down if no PVCs	Indicates whether the router considers the interface unavailable when there are no active PVCs.
Line speed (bps)	Indicates the configured data rate of the Frame Relay interface.
Timer T1 seconds	Indicates the frequency with which the Frame Relay interface performs a sequence number exchange with the Frame Relay switch LMI entity.
Counter N1 increments	Indicates the number of T1 timer intervals which must expire before a complete PVC LMI status enquiry is made.
LMI N2 error threshold	Indicates the number of management event errors occurring within the N3 window that will cause a reset of the Frame Relay interface.
LMI N3 error threshold window	Indicates the number of monitored management events used to measure the N2 error threshold.

MIR % of CIR	Minimum IR, expressed as a percentage of CIR.
IR % Increment	Percentage by which the router increments the IR each time it receives a frame without BECN until it reaches the maximum IR.
IR % Decrement	Percentage by which the router decrements the IR each time it receives a frame that contains BECN until it reaches the minimum IR.
DECnet length field	Indicates whether or not the DECnet length field feature is enabled. Some Frame Relay DECnet Phase IV implementations require a length field between the Frame Relay multiprotocol encapsulation header and the DECnet packet. A length field is inserted if the DECnet length field feature is enabled.
Current receive sequence	Indicates the current receive sequence number that the Frame Relay interface has received from the Frame Relay management entity.
Current transmit sequence	Indicates the current transmit sequence number that the Frame Relay interface has sent to the Frame Relay management entity.
Total status enquiries	Indicates the total number of status enquiries that the Frame Relay interface has made of the Frame Relay management entity.
Total status responses	Indicates the total number of responses that the Frame Relay interface has received from the Frame Relay management entity in response to status enquiries.
Total sequence requests	Indicates the total number of sequence number requests that the Frame Relay interface has sent to the Frame Relay management entity.
Total sequence responses	Indicates the total number of sequence number responses that the Frame Relay interface has received from the Frame Relay management entity.

PVC Status:

Total allowed	Indicates the number of allowable PVCs (including orphans) whose state is active, congested, removed, or inactive for use with this interface.
Total configured	Indicates the total number of currently configured PVCs for this interface.
Total active	Indicates the number of active PVCs on this interface.
Total congested	Indicates the number of PVCs that are throttled down because of congestion within the network.

> *Total left net*   Indicates the total number of PVCs that have been removed from the network.
>
> *Total join net*   Indicates the total number of PVCs that have been added to the network.

permanent-virtual-circuit

> Displays general link-layer statistics and configuration information for all configured PVCs on the Frame Relay interface.
>
> **Example:**   `list permanent-virtual-circuit`

```
                         Orphan  Type/   Frames       Frames
Circuit#    Circuit Name Circuit State Transmitted   Received
--------  ---------------- ------- ----- ----------- ---------
   16     Valencia         No    *P/A          2         1
   17     Raleigh          No    #P/A         15        14
   21     New York         No     P/A          2         0

  A - Active    I - Inactive   R - Removed   P - Permanent  C - Congested
  * - Required                    # - Required and belongs to a PVC group
```

> *Circuit#*   Indicates the number of the PVC.
>
> *Circuit Name*   Name of the circuit, an ASCII string.
>
> *Orphan Circuit*   Indicates whether the PVC is a nonconfigured circuit (yes or no).
>
> *Type/State*   Indicates the state of the circuit, A (active), I (inactive), P (permanent), C (congested), or R (removed).
>
> *Frames Transmitted*   Indicates how many frames this PVC has transmitted.
>
> *Frames Received*   Indicates how many frames this PVC has received.

pvc-groups

> Displays required PVC group information for all required PVC groups. For each group this consists of the group name, the circuits in the group and the state (active, inactive, or removed) of each circuit.
>
> **Example:**   `list pvc-groups`

```
          Group name          Circuits in group   Circuit status
          ----------          -----------------   --------------

          group1                     16                active
                                     44                inactive
                                    240                removed
```

LLC

Use the **LLC** command to access the LLC monitoring prompt. LLC commands are entered at this new prompt. See "LLC Monitoring Commands" on page 20-1 for an explanation of each of these commands.

Syntax: <u>ll</u>c

Example: `llc`

```
Circuit number to monitor [0]?
LLC user monitoring
LLC>
```

Note: The LLC command is supported only if APPN is in the software load.

Set

Use the **set** command to set the values for Committed Information Rate (CIR), Committed Burst Rate, and Excess Burst Rate for the specified PVC. You also can set values for IR adjustment rates.

Changes made with this command do not affect the configuration data, they are in effect only until the router is restarted.

Syntax: s̲et c̲ircuit . . .

i̲r-adjustment . . .

circuit

Sets the values for Committed Information Rate (CIR), Committed Burst Rate, and Excess Burst Rate for the specified PVC.

Syntax: s̲et c̲ircuit . . .

i̲r-adjustment . . .

Example: `set circuit`

```
Circuit number [16]?
Committed Information Rate (CIR) in bps [1200]?
Committed Burst Size (Bc) in bits [1200]?
Excess Burst Size (Be) in bits [56000]?
```

Circuit Number   Indicates the circuit number in the range 16 to 1007.

Committed Information Rate

Indicates the committed information rate (CIR). The CIR can be either 0 or a value in the range 300 bps to 2048000 bps. The default is 64000 bps. For more information, see "Committed Information Rate (CIR)" on page 26-9.

Committed Burst Size   The maximum amount of data in bits that the network agrees to deliver during a measurement interval equal to committed burst (Bc) size / CIR seconds. The range is 300 to 2048000 bits. The default value is 64000 bits.

Note: If CIR is configured as 0 then the committed burst size is set to 0 and you are not prompted for a value. For additional information, see "Committed Burst (Bc) Size" on page 26-9.

Excess Burst Size   The maximum amount of uncommitted data in bits in excess of committed burst size that the network attempts to deliver during a measurement interval equal to (Committed Burst Size/CIR) seconds. Range is 0 to 2048000 bits. Default is 0. For additional information, see "Excess Burst (Be) Size" on page 26-10.

ir-adjustment   *increment-% decrement-% minimum-IR*

Sets the minimum information rate (IR) and the percentages for incrementing and decrementing the IR in response to network congestion.

The minimum IR, expressed as a percentage of CIR, is the lower limit of the information rate. The minimum percentage is 1 and the maximum percentage is 100. The default is 25.

When network congestion clears, the information rate is gradually incremented by the IR adjustment increment percentage until the maximum information rate is reached. The minimum percentage is 1 and the maximum percentage is 100. The default is 12.

When network congestion occurs, the information rate is decremented by the IR adjustment decrement percentage each time a frame containing BECN is received until the minimum information rate is reached. The minimum percentage is 1, and the maximum percentage is 100. The default is 25.

Example: `set ir-adjustment`

```
IR adjustment % increment [12]?
IR adjustment % decrement [25]?
Minimum IR as % of CIR [25]?
```

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: `exit`

Frame Relay Interfaces and the GWCON Interface Command

While Frame Relay interfaces have a console process for monitoring purposes, the router also displays complete statistics for installed interfaces when you use the **interface** command from the GWCON environment. (For more information on the **interface** command, refer to Chapter 6, "The GWCON (Monitoring) Process and Commands")

Statistics Displayed For Frame Relay Interfaces

Statistics similar to the following are displayed when you execute the **interface** command from the GWCON environment for Frame Relay interfaces:

```
+interface 10
                                       Self-Test  Self-Test  Maintenance
Nt Nt'  Interface  Slot-Port           Passed     Failed     Failed
10 10   FR/0       Slot: 8  Port: 0       2          1          0

  Frame Relay MAC/data-link on V.35/V.36 interface

  Adapter cable:              V.35 DTE

  V.24 circuit: 105 106 107 108 109
  Nicknames:    RTS CTS DSR DTR DCD
  PUB 41450:    CA  CB  CC  CD  CF
  State:        ON  ON  ON  ON  ON

  Line speed:              64.000 Kbps
  Last port reset:        1 hour, 20 minutes, 42 seconds ago

  Input frame errors:
   CRC error                    0    alignment (byte length)        0
   missed frame               182    too long (>  2062 bytes)       0
   aborted frame                0    DMA/FIFO overrun               0
  Output frame counters:
   DMA/FIFO underrun errors      0   Output aborts sent             0
```

Nt   Indicates the interface number as assigned by software during initial configuration.

Nt'	Indicates the interface number as assigned by software during initial configuration.
	Note: For FR dial circuit interfaces, Nt' is different from Nt. Nt' indicates the base interface (ISDN) that the dial circuit is running over.
Interface	Indicates the type of interface and its instance number. Frame relay has a FR designation.
Slot	Indicates the slot of the interface running Frame Relay
Port	Indicates the port of the interface that is running Frame Relay
Self-test Passed	Indicates the total number of times the Frame Relay interface passed self-test.
Self-test Failed	Indicates the total number of times the Frame Relay interface failed self-test.
Maintenance Failed	Indicates the total number of times the interface was unable to communicate with Frame Relay management.
V.24 circuit, Nicknames, and State	
	The circuits, control signals, pin assignments and their state (ON or OFF). Note: The symbol - - - in console output indicates that the value or state is unknown.
Line speed	The transmit clock rate.
Last port reset	The length of time since the last port reset.
Input frame errors:	
CRC error	The number of packets received that contained checksum errors and as a result were discarded.
Alignment	The number of packets received that were not an even multiple of 8 bits in length and a result were discarded.
Too short	The number of packets that were less than 2 bytes in length and as a result were discarded.
Too long	The number of packets that were greater than the configured size, and as a result were discarded.
Aborted frame	The number of packets received that were aborted by the sender or a line error.
DMA/FIFO overrun	The number of times the serial interface could not send data fast enough to the system packet buffer memory to receive them from the network.
Missed frame	When a frame arrives at the device and there is no buffer available, the hardware drops the frame and increments the missed frame counter.
L & F bits not set	On serial interfaces, the hardware sets input-descriptor information for arriving frames. If the buffer can accept the complete frame upon arrival, the hardware sets both the last and first bits of the frame, indicating that the buffer accepted the complete frame. If either of the bits is not

set, the packet is dropped, the `L & F bits not set` counter is incremented, and the buffer is cleared for reuse.

Note: It is unlikely that the `L & F bits not set` counter will be affected by traffic.

Output frame counters:

DMA/FIFO underrun errors

The number of times the serial interface could not retrieve data fast enough from the system packet buffer memory to transmit them to the network.

Output aborts sent The number of transmissions that were aborted as requested by upper-level software.

Statistics similar to the following are displayed for Frame Relay dial circuits when you execute the interface command from the GWCON environment:

```
+interface 3
                            Self-Test  Self-Test  Maintenance
                                       Self-Test  Self-Test  Maintenance
Nt Nt'  Interface           Passed     Failed     Failed
3  2    FR/1                    1          0          0

  Frame Relay MAC/data-link on ISDN Primary Rate interface
```

Chapter 28. Using and Configuring Point-to-Point Protocol Interfaces

This chapter describes how to configure the Point-to-Point Protocol for interfaces on the device. Sections in this chapter include:

- "PPP Overview"
- "The PPP Link Control Protocol (LCP)" on page 28-3
- "The PPP Network Control Protocols" on page 28-9
- "PPP Authentication Protocols" on page 28-7
- "Accessing the Interface Configuration Process" on page 28-11
- "Point-to-Point Configuration Commands" on page 28-11

PPP Overview

Figure 28-1 shows some examples of point-to-point serial links.
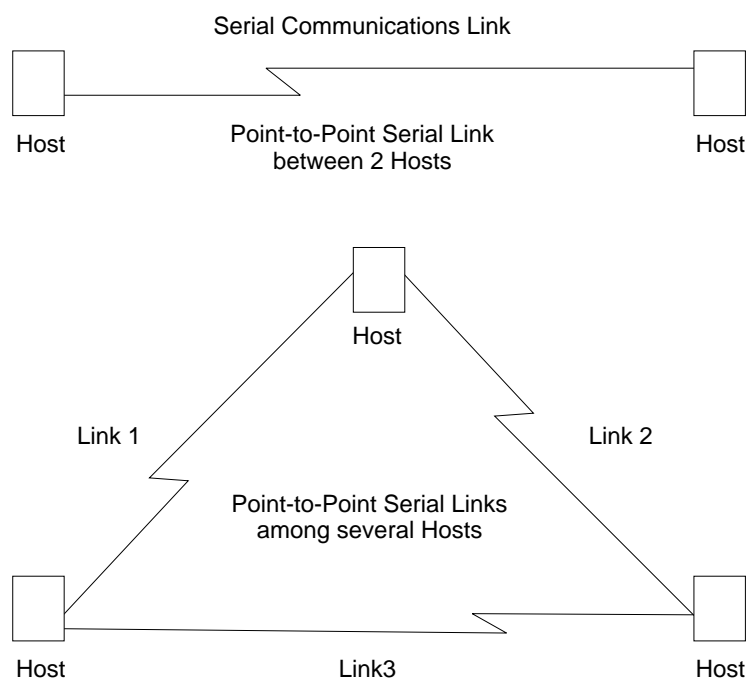
Figure 28-1. Examples of Point-to-Point Links

PPP provides a method for transmitting protocol datagrams at the Data Link Layer over serial point-to-point links. PPP supports synchronous data transmission and provides the following services:

- Link Control Protocol (LCP) to establish, configure, and test the link connection.

- Encapsulation protocol for encapsulating protocol datagrams (through the HDLC method) over serial point-to-point links.

- Authentication protocols (AP) to validate the identity of a peer (remote) unit, and to submit your own identity to the peer for validation.

- Network Control Protocols (NCP) for establishing and configuring different network layer protocols. PPP allows the use of multiple network layer protocols.

PPP currently supports AppleTalk** Control Protocol (ATCP), DECnet** Protocol Control Protocol (DNCP), Banyan** VINES** Control Protocol (BVCP), Bridging Control Protocol (BCP), Internet Protocol Control Protocol (IPCP), IPX Control Protocol (IPXCP) APPN HPR Control Protocol (APPN HPRCP), APPN ISR Control Protocol (APPN ISRCP), and OSI Control Protocol (OSICP).

Each end starts by sending LCP packets to configure and test the data link. After the link has been established, PPP sends NCP packets to choose and configure one or more network layer protocols. After network layer protocols have been configured, datagrams from each network layer can be sent over the link. The next sections explain these concepts in more detail.

PPP Data Link Layer Frame Structure

PPP transmits data frames that have the same structure as High-level Data Link Control (HDLC) frames. PPP uses a byte-oriented transmission method with a single-frame format for all data and control exchanges. Figure 28-2 illustrates the PPP frame structure and is followed by a detailed description of each field. PPP links can negotiate the use of various options which may modify the basic frame format; the description below applies to the frame format prior to any such modifications. PPP LCP packets are always sent in this format as well, regardless of negotiated options, so that LCP packets can be recognized even when there is a loss of synchronization on the line.

The router supports two such options: Address and Control Field Compression (ACFC) and Protocol Field Compression (PFC). These are described in detail in a later section.

Flag	Address	Control	Protocol	Information	FCS	Flag
8 bits	8 bits	8 bits	16 bits	variable	16 bits	8 bits

Figure 28-2. PPP Frame Structure

Flag Fields

The flag field begins and ends each frame with a unique pattern of 01111110. Generally a single flag ends one frame and begins the next. The receiver attached to the link continuously search for the flag sequence to synchronize the start of the next frame. The sending station and the receiving station use a process called bit stuffing to avoid the loss of synchronization due to the arbitrary appearance of the 01111110 bit pattern within the data stream.

When bit stuffing, the sending station adds a 0 after the fifth 1 in a non-flag bit stream. Then the receiving station monitors the bit stream. If the receiver detects a bit pattern of five ones, it examines the sixth bit. If the sixth bit is a 0, the receiver deletes it. If the sixth bit is a 1 and the seventh bit is a 0, the receiver accepts the combination as the end of frame flag.

Address Field

The address field is a single octet (8 bits) and contains the binary sequence 11111111 (0xff hexadecimal). This is known as the All-Station Address. PPP does not assign individual station addresses.

Control Field

The control field is a single octet and contains the binary sequence 00000011 (0x03 hexadecimal). This sequence identifies the Unnumbered Information (UI) command with the P/F bit set to zero.

Protocol Field

The protocol field is defined by PPP and is not found in the true HDLC frame format. The protocol field is 2 octets (16 bits) and its value identifies the protocol datagram encapsulated in the Information field of the frame.

Protocol field values in the range '0xC000'–'0xFFFF' indicate Layer 3 data (protocol datagrams) such as LCP, PAP, CHAP, and CCP. Values in the range '8000'–'BFFF' indicate that the datagrams belong to the Network Control Protocols (NCP). Values in the range '0'–'3FFF' identify the network protocol of specific datagrams.

Information Field

The information field contains the datagram for the protocol specified in the protocol field. This is zero or more octets. This information field is present for all I-frames and only some UI-frames (unnumbered I-frames).

When the protocol type is LCP, exactly one LCP packet is encapsulated in the information field of PPP Data Link Layer frames.

Frame Check Sequence (FCS) Field

The frame check sequence field is a 16-bit cyclic redundancy check (CRC).

The PPP Link Control Protocol (LCP)

PPP's Link Control Protocol (LCP) establishes, configures, maintains, and terminates the point-to-point link. This process is carried out in four phases:

1. Before exchanging any network layer datagrams, PPP first opens the connection through an exchange of LCP configuration packets. As part of this negotiation process, the PPP processes at each end of the link agree on various basic link level parameters such as the maximum packet size that can be transferred and whether the ends must use an authentication mechanism to identify themselves to their peers before carrying network traffic.

   If this negotiation is unsuccessful, the link is considered to be "down" and incapable of carrying any network traffic. If the negotiation is successful, LCP goes to an "Open" state and PPP goes on to the next phase.

2. After LCP successfully reaches an Open state, the next step in establishing the link is to perform authentication where each end of the link identifies itself to the other end using the "authentication protocol" that the other end dictated as part of the LCP negotiation.

   If authentication fails, the link is marked "down" and cannot carry any network traffic. If authentication succeeds or if authentication is not required, the PPP link moves to the next phase, NCP negotiation. The link is considered to be "open" or "up" at this time, though it cannot yet carry layer-3 protocol datagrams.

3. Once the link is open, the router negotiates the use of various layer-3 protocols (for example, IP, IPX, DECnet, Banyan Vines) using Network Control Protocol (NCP) packets. Each layer-3 protocol has its own associated network control protocol. For example IP has IPCP and IPX has IPXCP. The basic format and

mechanisms for all these NCP packets is the same for all protocols, and is basically a superset of the LCP mechanisms as described later in this section.

Each layer-3 protocol is negotiated independently. When a particular NCP successfully negotiates, the link is "up" for that protocol's traffic. As with LCP, configuration information can be exchanged as part of this negotiation; for example, IPCP can exchange IP addresses or negotiate the use of "Van Jacobson IP header compression".

As with LCP, it is possible for an NCP to fail to negotiate successfully with its peer. This might be because the peer does not support a particular protocol or because some configuration option was unacceptable. If an NCP fails to reach the "Open" state, no layer-3 protocol packets can be exchanged for that protocol even though other layer-3 protocols are successfully passing traffic across the PPP link.

4. Finally, LCP has the ability to terminate the link at any time. This is usually done at the request of the user but may occur because of the loss of a carrier or the expiration of an idle-period timer.

For complete details about PPP LCP, authentication, and the general NCP negotiation mechanisms, consult RFCs 1331, 1334, 1570, and 1661.

LCP Packets

LCP packets are used to establish and manage a PPP link and can be loosely divided into three categories:

- *Link establishment packets* that exchange configuration information and establish the link.

- *Link termination packets* that shut down the link or signal that a link is not accepting connections at a particular time. They also can be used to signal that a particular protocol is unrecognized (for example, during NCP negotiations).

- *Link maintenance packets* that monitor and debug a link.

Exactly one LCP packet is encapsulated in the information field of PPP Data Link Layer frames. In the case of LCP packets, the protocol field reads "Link Control Protocol" (C021 hexadecimal). Figure 28-3 illustrates the structure of the LCP packet and is followed by a detailed description of each field.

| Code | Identifier | Length | Data(option) |

Figure 28-3. LCP Frame Structure (in PPP Information Field)

Code

The code field is one octet in length and identifies the type of LCP packet. The codes in Table 28-1 on page 28-5 distinguish the packet types. They are described in more detail in later sections.

Table 28-1. LCP Packet Codes	
Code	**Packet Type**
1	Configure-Request (Link Establishment)
2	Configure-Ack (Link Establishment)
3	Configure-Nak (Link Establishment)
4	Configure-Reject (Link Establishment)
5	Terminate-Request (Link Termination)
6	Terminate-Ack (Link Termination)
7	Code-Reject (Link Establishment)
8	Protocol-Reject (Link Establishment)
9	Echo-Request (Link Maintenance)
10	Echo-Reply (Link Maintenance)
11	Discard-Request (Link Maintenance)

Identifier

The identifier field is one octet in length and is used to match packet requests and replies.

Length

The length field is two octets in length and indicates the total length (that is, including all fields) of the LCP packet.

Data (Option)

The data field is zero or more octets as indicated by the length field. The format of this field is determined by the code.

NCP packets are structured identically to LCP packets and are distinguished by having different PPP "Protocol" values. Each LCP packet type (distinguished by the code field) has the same meaning for each NCP, though an individual NCP may not implement all possible LCP packet types. NCPs normally implement all of the link establishment type packets that LCP defines. They may implement some of the additional LCP packet types, and they also may define additional packet types beyond what LCP uses. Unlike LCP packets, the structure of an NCP frame may be modified according to options negotiated by LCP during the link establishment phase.

Link Establishment Packets

Link Establishment Packets establish and configure a point-to-point link including the following packet types:

Configure-Request

LCP packet code field is set to 1. LCP transmits this packet type when it wants to open a point-to-point link. Upon receiving a Configure-Request, a peer station's LCP entity must send an appropriate reply.

Configure-Ack

LCP packet code field is set to 2. The peer transmits this packet type when every configuration option in a Configure-Request packet is acceptable. Upon receiving the Configure-Ack (ack = acknowledgement), the originating station checks the Identifier field. This field must match the one from the last-transmitted Configure-Request or the packet is invalid.

> Both ends send Configure-Request and both ends must receive a
> Configure-Ack before the link opens. Options negotiated for one direction
> may differ from that negotiated for the other direction. There is no
> "master-slave" relationship. Rather, each end works symmetrically.

Configure-Nak
> LCP packet code field is set to 3. The peer transmits this packet type when
> some part of the configuration option in a Configure-Request packet is
> unacceptable. The Identifier field is copied from the received
> Configure-Request and the Data (option) field is filled with the received
> unacceptable configuration options. The Identifier field must match the one
> from the last-transmitted Configure-Request or the packet is invalid and is
> discarded.
>
> When the originator receives a Configure-Nak packet, a new
> Configure-Request packet is sent that includes modified, acceptable
> configuration options.

Configure-Reject
> LCP packet code field is set to 4. The peer transmits this packet type when
> some part of the configuration options in a Configure-Request packet is
> unacceptable. The Identifier field is copied from the received
> Configure-Request and the Data (option) field is filled with the received
> unacceptable configuration options. The Identifier field must match the one
> from the last-transmitted Configure-Request or the packet is invalid and is
> discarded.
>
> When the originator receives a Configure-Reject packet, a new
> Configure-Request packet is sent that does not include any of the
> configuration options received in the Configure-Reject packet.

Code-Reject
> LCP packet code field is set to 7. The transmission of this packet type
> indicates that the LCP "code" field on a received packet is not recognized as
> a valid value. While this can indicate an error, it also can indicate that the
> peer does not implement some feature that you are trying to use.

Protocol-Reject
> LCP packet code field is set to 8. The transmission of this packet type
> indicates that a PPP frame has been received that contains an unsupported
> or unknown protocol (the PPP "protocol" field was unrecognized for some
> packet). This usually occurs if you try to negotiate some NCP for a protocol
> that the other end doesn't support. For example, if DECNet CP (DNCP)
> sends a Config-Request and the other end does not know about DECNet, the
> other end replies with an LCP Protocol-Reject on DNCP. Upon receiving a
> Protocol-Reject packet, the link stops transmitting the incorrect protocol.

Note: NCP packet types and structure are the same as LCP, although there are a
few additional "code" fields associated with some NCPs.

Link Termination Packets

Link Termination Packets terminate a link and include the following packet types:

Terminate-Request
> LCP packet code field is set to 5. LCP transmits this packet type when a
> point-to-point link needs to be closed. These packets are sent until a
> Terminate-Ack packet is sent back, or until a retry counter is exceeded while
> waiting for an Ack.

Terminate-Ack
> LCP packet code field is set to 6. Upon receiving a Terminate-Request packet, this packet type must be transmitted with the code field set to 6. Reception of an Terminate-Ack packet that was not expected indicates that the link has been closed.

Link Maintenance Packets

Link Maintenance Packets manage and debug a link, and include the following packet types:

Echo-Request and Echo-Reply
> LCP packet code fields are set to 9 and 10 respectively. LCP transmits these packet types in order to provide a Data Link Layer loopback mechanism for both directions on the link. This feature is useful, for example, in debugging a faulty link to determine link quality. These packets are sent only when the link is in the Open state.

Discard-Request
> LCP packet code field is set to 11. LCP transmits this packet type to provide a data sink for Data link Layer testing. A peer that receives a Discard-Request *must* throw away the packet. This is useful in debugging a link. These packets are sent only when the link is in the Open state.

PPP Authentication Protocols

PPP authentication protocols provide a form of security between two nodes connected via a PPP link. If authentication is required on a box, then immediately after the two boxes successfully negotiate the use of the link at the LCP layer (LCP packets are exchanged until LCP goes into an "open" state), they go into an "authentication" phase where they exchange authentication packets. A box is neither able to carry network data packets nor negotiate the use of a network protocol (NCP traffic) until authentication negotiation completes successfully.

There are different authentication protocols in use, PAP (Password Authentication Protocol) and CHAP (Challenge/Handshake Authentication Protocol). These are described in detail in RFC 1334, and briefly described later in this section.

Whether a box requires the other end to authenticate itself (and if so, with what protocol) is determined during the LCP negotiation phase. Hence, in some sense authentication could be considered to "fail" even at the link establishment phase (LCP negotiation), if one end does not know how, or refuses, to use the authentication protocol the other end requires.

Each end of a link sets its own requirements for how it wants the other end to authenticate itself. For example, given two routers "A" and "B," connected over a PPP link, side A may require that B authenticate itself to A using PAP, and side B may require that A similarly identify itself using CHAP. It is valid for one end to require authentication while the other end requires none.

In addition to initial authentication during link establishment, with some protocols an authenticator may demand that the peer reestablish its credentials periodically. With CHAP, for example, a rechallenge may be issued at any time by the authenticator and the peer must successfully reply - or lose the link.

If more than one authentication protocol is enabled, the router initially attempts to use them in the priority order:

1. CHAP
2. PAP.

If the remote side responds to the authentication request with NAK and suggests an alternative, the router uses the alternative provided it is enabled on the link. If the remote side continues responding to the router's suggestions with a NAK but does not provide an alternative that the router has enabled, the link is terminated.

Password Authentication Protocol (PAP)

The Password Authentication Protocol (PAP) provides a simple method for the peer to establish its identity using a two-way handshake. This is done only upon initial link establishment. Following link establishment, the peer sends an ID/Password pair to the authenticator until authentication is acknowledged or the connection is terminated. Passwords are sent over the circuit "in the clear," and there is no protection from playback or repeated trial and error attacks. The peer controls the frequency and timing of the attempts.

Challenge-Handshake Authentication Protocol (CHAP)

The Challenge-Handshake Authentication Protocol (CHAP) is used to periodically verify the identity of the peer using a three-way handshake. This is done upon initial link establishment, and *may* be repeated anytime after the link has been established. After the initial link establishment, the authenticator sends a "challenge" message to the peer. The peer responds with a value calculated using a "one-way hash" function. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise the connection is terminated.

Configuring Authentication

The following paragraphs describe configuring PPP authentications for two situations:

- Configuring IBM 2216 to authenticate a remote router or device
- Configuring IBM 2216 to be authenticated by a remote router or device.

These two situations are *independent*. You can do one or the other or both.

Configuring IBM 2216 to Authenticate Remote Router or Device

To configure the IBM 2216 to authenticate a remote router or device, do the following:

1. Enable authentication on the PPP interface

   - At the Config> prompt, enter the network command for the PPP interface that the remote router or device is attached to.

   - At the PPP Config> prompt, enable the authentication protocol that you want to use by typing either "enable pap" or "enable chap".

2. Enter the name and password of the remote router or device into the PPP user database.

   - At the Config> prompt, use the "add ppp_user" command.

The IBM 2216 maintains a single PPP user database. When the remote router or device sends its name and password to the IBM 2216 during the authentication phase, the IBM 2216 checks to see if that name and password are in the PPP user database.

Configuring IBM 2216 to be Authenticated by Remote Router or Device

To configure the IBM 2216 to be authenticated by a remote router or device, configure the IBM 2216's name and password:

1. At the Config> prompt, enter the network command for the PPP interface that will be authenticated by a remote router or device.

2. At the PPP Config> prompt, type the "set name" command and provide the name and password that the IBM 2216 will use to identify itself to the remote router or device during the authentication phase.

Attention: Do not use the "enable pap" or "enable chap" commands unless you want the IBM 2216 to authenticate the remote router or device as described in "Configuring IBM 2216 to Authenticate Remote Router or Device" on page 28-8.

The PPP Network Control Protocols

PPP has a family of Network Control Protocols (NCPs) for establishing and configuring different network layer protocols. The NCPs are responsible for configuring, enabling, and disabling the network layer protocols on both ends of the point-to-point link. NCP packets cannot be exchanged until LCP has opened the connection and the link reaches the OPEN state.

PPP supports the following Network Control Protocols:

- AppleTalk Control Protocol (ATCP)
- Banyan VINES Control Protocol (BVCP)
- Bridging Control Protocol (BCP),
- DECnet Control Protocol (DNCP)
- IP Control Protocol (IPCP)
- IPX Control Protocol (IPXCP)
- OSI Control Protocol (OSICP)
- APPN High Performance Routing Control Protocol (APPN HPRCP)
- APPN Intermediate Session Routing Control Protocol (APPN ISRCP)

AppleTalk Control Protocol

ATCP is specified in Request for Comments (RFC) 1378. IBM's implementation of ATCP supports the AppleTalk-Address option. The implementation supports both full router mode and half router mode. For additional information, refer to "AppleTalk over PPP" in *Protocol Configuration and Monitoring Reference Volume 2 for Nways Multiprotocol Access Services Version 1 Release 1*.

Banyan VINES Control Protocol

RFC 1763 describes BVCP. IBM's implementation of BVCP does not support any options.

Bridging Control Protocol

BCP is specified in RFC 1220. IBM's implementation of BCP supports IEEE 802.5 Line Identification Option and the Tinygram Compression Option.

DECnet Control Protocol

DNCP is specified in RFC 1376. IBM's implementation does not support any DNCP options.

IP Control Protocol

IPCP is specified in RFC 1332. IBM's implementation supports the IPCP options:

- IP Compression Protocol

  The only IP compression protocol supported is the Van Jacobson compression protocol as described in RFC 1144.

- IP Address

  The router can query the remote end for its IP address and can supply its own address to the peer. The router does not support dynamic assignment of IP addresses. It neither assigns an address to a remote unit that requests one, nor accepts a request from the peer to change its own port address.

IPX Control Protocol

IPXCP is specified in RFC 1552. IBM's implementation does not support any IPXCP options.

OSI Control Protocol

OSICP is specified in RFC 1377. IBM's implementation of OSICP does not support any options.

APPN HPR Control Protocol

Advanced Peer-to-Peer Networking (APPN) High Performance Routing (HPR) control protocol is specified in RFC 2043. No options are negotiated for this control protocol.

APPN ISR Control Protocol

Advanced Peer-to-Peer Networking (APPN) Intermediate Session Routing (ISR) control protocol is specified in RFC 2043. No options are negotiated for this control protocol.

Accessing the Interface Configuration Process

Use the following procedure to access the router's configuration process. This process gives you access to a specific interface's *configuration* process.

1. At the OPCON prompt (*), enter the **status** command to find the PID for CONFIG. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter the OPCON **talk** command and the PID for CONFIG. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

   `* talk 6`

   After you enter the talk 6 command, the CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter **CONFIG**, press **Return** again.

3. At the CONFIG prompt, enter the **list devices** command to display the network interface numbers for which the router is currently configured.

4. Record the interface numbers.

5. Enter the CONFIG **network** command and the number of the interface you want to configure. For example:

   `Config> network 1`

   The appropriate configuration prompt (such as `TKR Config>` for token-ring), now displays on the console.

Note: Not all network interfaces are user-configurable. For interfaces that cannot be configured, you receive the message:

   `That network is not configurable`

Displaying the PPP Configuration Prompt

To display the `PPP config>` prompt:

1. Enter **list devices** at the `Config>` prompt to display a list of interfaces.

2. If you have not already done so, set the data link protocol on one of the serial interfaces to PPP by entering **set data-link ppp** at the `Config>` prompt. For example:

   ```
   Config> set data-link ppp
   Interface Number [0]? 2
   ```

3. Enter **network** followed by the number of the PPP interface. For example:

   ```
   Config> network 2
   PPP config>
   ```

Point-to-Point Configuration Commands

Table 28-2 on page 28-12 summarizes the PPP configuration commands, and the rest of this section explains these commands. Enter the commands at the `PPP config>` prompt.

Table 28-2. Point-to-Point Configuration Command Summary	
Command	**Function**
? (Help)	Displays all the Point-to-Point commands or lists the options for specific commands (if available).
Disable	Disables data compression (CCP), DTR line handling, CHAP, PAP.
Enable	Enables data compression (CCP), DTR line handling, CHAP, PAP.
List	Lists all information related to the point-to-point interfaces protocols, parameters, and options.
Set	Sets physical line (HDLC) parameters, LCP parameters, generic NCP parameters, and various NCP-specific options.
Exit	Returns to the `Config>` prompt.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
DISABLE
ENABLE
LIST
SET
EXIT
```

Example: **list ?**

```
ALL
BCP
CCP
HDLC
IPCP
LCP
NCP
```

Disable

Disables data compression, authentication protocols, and the Lower DTR feature.

Syntax: <u>d</u>isable <u>cc</u>p
                     <u>ch</u>ap
                     <u>l</u>ower-dtr
                     <u>p</u>ap

ccp
> Disables the use of data compression on the interface. Refer to Chapter 30, "Data Compression for PPP Interface" on page 30-1 for more information.

chap
> Disables the use of the Challenge-Handshake Authentication Protocol. Refer to "Challenge-Handshake Authentication Protocol (CHAP)" on page 28-8 for more information.

lower-dtr
> Determines the way the data terminal ready (DTR) signal is handled for leased serial-line interfaces that are disabled. If this parameter is set to

"disabled" (the default), the DTR signal will be raised when the interface is disabled.

pap

Disables the use of the Password Authentication Protocol. Refer to "Password Authentication Protocol (PAP)" on page 28-8 for more information.

Enable

Enables data compression, authentication protocols, including PAP and CHAP, and lower-DTR on this PPP interface. If multiple authentication protocols are enabled, the router attempts to use them in the following priority order:

1. CHAP
2. PAP

Syntax: <u>en</u>able <u>cc</u>p
<u>ch</u>ap
<u>l</u>ower-dtr
<u>p</u>ap

ccp

Enables the use of data compression on the interface. Refer to the PPP data compression chapter for more details on data compression.

chap

Enables the use of the Challenge-Handshake Authentication Protocol. You are prompted for a rechallenge interval. Specify 0 if you do not want to rechallenge periodically after the initial authentication phase is complete. Refer to "Challenge-Handshake Authentication Protocol (CHAP)" on page 28-8 for more information.

Syntax: <u>e</u>nable <u>ch</u>ap

```
Rechallenge Interval in seconds (0=NONE)  [0] 10
CHAP enabled
```

lower-dtr

Determines the way the data terminal ready (DTR) signal is handled for leased serial-line interfaces that are disabled. If this parameter is set to "disabled" (the default), the DTR signal will be raised when the interface is disabled.

If Lower DTR is set to "enabled," then the DTR signal will be lowered when the interface is disabled. This behavior may be desirable in situations where the interface has been configured as an alternate link for WAN Reroute and the interface is connected to a dial-out modem which maintains its dial connection based on the state of the DTR signal.

When the interface is disabled, the DTR signal is low and the modem keeps the dial connection down. When the interface is enabled, due to a WAN Reroute backup scenario, DTR is raised and the modem dials a stored number to the backup site. When the primary interface is restored, the alternate interface is disabled, DTR is lowered, and the modem hangs up the dial connection.

The following cable types are supported:

RS-232
V.35
V.36

> **Note:** The **enable lower-dtr** command is not supported on PPP dial circuit interfaces.

pap
> Enables the use of the Password Authentication Protocol. Refer to "Password Authentication Protocol (PAP)" on page 28-8 for more information.

List

Use the **list** command to display information related to the PPP interface and its protocol parameters and options.

On PPP dial circuit interfaces, the "list hdlc" option is not available. For dial circuits, hardware data link parameters are a function of the base net rather than the PPP dial circuit. For additional information, see Chapter 39, "Configuring Dial Circuits" on page 39-1.

Syntax: list　　all
　　　　　　　　bcp
　　　　　　　　ccp
　　　　　　　　hdlc
　　　　　　　　ipcp
　　　　　　　　lcp
　　　　　　　　ncp

all
> Lists all options and parameters related to the PPP interface.
>
> The **list all** command displays the output of *all* the individual **list...** parameters described below.

bcp
> Lists the Bridging Network control protocol options.
>
> **Example:**　**list bcp**
>
> ```
> BCP Options
> ------------
> Tinygram Compression:DISABLED
> ```

Tinygram Compression:
> Displays whether Tinygram Compression is enabled/disabled.

ccp
> Displays the currently selected data compression options.
>
> **Example:**　**list ccp**
>
> ```
> CCP Options
> ------------
> Data Compression disabled
> Algorithm list: STAC-LZS
> Stac: histories 1
> Stac: check_mode SEQ
> ```

Data Compression Enabled/Disabled
> Indicates whether data compression is enabled or disabled on this interface. For additional information, see Chapter 30, "Data Compression for PPP Interface" on page 30-1.

Algorithm list
> Displays the current compression algorithms followed by the options configured using the **set ccp options** command.

> **Note:** The only compression scheme supported is STAC-LZS, described in RFC 1974.

Stac: histories
> The number of separate compression histories to maintain.

Stac: check_mode
> The method used to detect lost incoming packets.

hdlc
> Lists parameters related to the High-Level Data Link Control (HDLC) protocol.
>
> **Note:** This command is not supported on PPP dial circuit interfaces.
>
> The Lower DTR state is displayed only if the cable type is not X.21.
>
> **Example:** `list hdlc`
>
> ```
> Encoding: NRZ
> Idle State: Flag
> Clocking:  Internal
> Cable type:  V.35 DCE
> Speed (bps): 6400
>
> Transmit Delay Counter:  0
> Lower DTR:  Disabled
> ```

Encoding:
> HDLC transmission encoding scheme, either NRZ (non-return to zero) or NRZI (non-return to zero inverted).

Idle State:
> Bit pattern, either Flag or Mark, transmitted on the point-to-point link when the interface is not transmitting data.

Clocking:
> Interface clocking, either external or internal.

Cable type:
> Specifies the type of cable in use (RS-232, V.35, or V.36).

Speed (bps):
> The physical data rate of the interface. When clocking is internal, this is the data rate generated by the internal clock.

Transmit Delay Counter:
> Number of flags sent between frames.

Lower DTR:
> Enabled or Disabled. If Lower DTR is enabled, the router drops the DTR signal when a WAN Reroute alternate link is no longer needed. Dropping the DTR signal causes the modem to terminate the leased-line connection for the alternate link.

ipcp
> Lists the Internet Protocol control protocol options.
>
> **Example:** `list ipcp`
>
> ```
> IPCP Options
> ------------
> IPCP Compression:      NONE
> IP Address:            Don't Send or Request
> ```

IPCP compression

Indicates whether the PPP handler accepts compressed IP data. PPP supports Van Jacobson TCP/IP header compression (RFC 1144). Enable this option when the point-to-point link is running at a low baud rate.

A value of "Van Jacobson" indicates that compression is supported. A value of "NONE" indicates that compressed data is not being accepted.

IP address

Indicates whether IPCP is configured to send the local IP address to the remote end of the link. Also displayed is whether IPCP is configured to request the IP address *from* the remote end of the link.

lcp

Lists the parameters and options for the Link Control Protocol.

Example: `list lcp`

```
LCP Parameters
--------------
Config Request Tries:        20   Config Nak Tries:          10
Terminate Tries:             10   Retry Timer:             3000

LCP Options
-----------
Max Receive Unit:          2048   Magic Number:             Yes
Protocol Field Comp (PFC)    No   Addr/Cntl Field Comp(ACFC) Yes

Authentication Options
-------------- -------
Authentication remote using:  none
Identify Self As              ibm
```

Config Request Tries:

Number of times that LCP sends configure-request packets to a peer station while attempting to open a PPP link.

Config Nak Tries:

Number of times that LCP sends configure-nak ("not acknowledged") packets to a peer station while attempting to open a PPP link.

Terminate Tries:

Number of times that LCP sends terminate-request packets to a peer station to close a PPP link.

Retry Timer:

Number of milliseconds that elapse before packet transmission continues according to the number of times set by the "Config tries" parameter.

Max Receive Unit:

Displays the maximum information field (packet) size handled by the link.

Magic Number:

Indicates whether the magic number loopback detection option is enabled.

Protocol Field Comp (PFC):

Indicates whether the PFC option is enabled.

Addr/Cntl Field Comp(ACFC):

Indicates whether ACFC is enabled.

Authentication remote using:

A list of enabled authentication protocols.

Identify Self As:

The name set with the **set name** command.

ncp

Lists the parameters for all Network Control Protocols.

Example: `list ncp`

```
NCP Parameters
---------------
Config Request Tries:    20   Config Nak Tries:      10
Terminate Tries:         10   Retry Timer:         3000
```

Config Request Tries:

Number of times NCP sends configure-request packets to a peer station while attempting to open a PPP link.

Terminate Tries:

While awaiting a Terminate-Ack, the number of times NCP sends Terminate-Request before it closes a PPP link.

Config Nak Tries:

Number of times NCP sends configure-nak (not acknowledged) packets to a peer station while attempting to open a PPP link.

Retry Timer:

Number of milliseconds that elapse before timing out of NCP's transmission of configure-request packets (to open the link) and terminate-request packets (to close the link).

LLC

Use the **LLC** command to access the LLC configuration environment (available only if APPN is included in the software load). See "LLC Configuration Commands" on page 19-1 for an explanation of each of these commands.

Syntax: <u>llc</u>

Example: `llc`

```
LLC config>
```

Set

Use the **set** command to set HDLC parameters, LCP options and parameters, IPCP options, BCP options, and NCP parameters. "Parameters" are related to internal operations for such things as retry counts. "Options" are things that are negotiated with the other end.

Note: Values immediately following the command option prompts reflect the current setting of that option. They are not always the default values illustrated in this chapter.

Syntax: <u>se</u>t

    <u>b</u>cp
    <u>c</u>cp <u>o</u>ptions
    <u>h</u>dlc...
    <u>i</u>pcp
    <u>l</u>cp...
    <u>n</u>ame
    <u>n</u>cp...

Note: The **set hdlc cable**, **set hdlc clocking**, **set hdlc encoding**, **set hdlc idle**, **set hdlc mode**, **set hdlc speed**, and **set hdlc transmit delay** commands are not supported on PPP dial circuit interfaces.

bcp
> Sets the Bridging Control Protocol (BCP) parameters.
>
> **Example:**   `set bcp`
>
> > `TINYGRAM COMPRESSION  [no]:`
>
> *Tinygram Compression*
> > Specifies whether or not Tinygram Compression is used. This option is useful for protocols that are prone to problems when bridged over low-speed (64 Kbps and below) lines. These protocols add zeroes between the data and the frame checksum to pad the Protocol Data Unit (PDU) to the minimum size. Tinygram compression removes the zeroes and preserves the frame checksum at the transmitting end. At the receiving end, it restores the packet to the minimum length.

ccp options
> Prompts you for the configurable options of the compression algorithms. Some of the options may be modified later by PPP negotiations with the peer router on the WAN link.
>
> **Example:**   `set ccp options`
>
> > `STAC: # histories [1]?`
> > `STAC: check mode (0=none, 1-LCB, 2=CRC, 3=Seq) [3]?`
>
> STAC: # histories
> > This sets the number of compression "contexts" or "histories" that are used by the STAC compression engine.
> >
> > A non-zero value means that the compression engine maintains a history where it keeps information about previous data sent in packets. This historical data is used to improve the effectiveness of the compression.
> >
> > The receiver maintains a similar history and as long as the transmitter and receiver keep their histories in sync, the receiver can properly decompress the packets it receives. If the histories get out of sync, packets are decompressed as unusable data. Normally, you should set the number of histories to 1 unless the link quality is very poor.
> >
> > A value of zero means that each packet sent is compressed without regard to any past packets sent and may always be reliably decompressed by the receiver. However, since the compressor cannot exploit any information derived from examining prior packets, the effectiveness of the compression usually is not as good.
> >
> > Some implementations support more than one history, subdividing the data stream into separate streams that are compressed independently. The router does not support the use of more than one history on a PPP link.
>
> STAC: check mode (0=none, 1=LCB, 2=CRC, 3=Seq)
> > STAC compressed datagrams normally include a check value used by the two ends of the link to recognize when a compressed packet has been lost or corrupted, and some action is needed to resynchronize the sender's and receiver's histories.
> >
> > This option sets the exact form of check value used. Choose one of the following:
> >
> > 0   None: No check value is used. Without a check value, there is no way to determine that a packet has been lost. Do not use this mode unless the underlying data link provides reliable, sequenced packet delivery.

1 LCB: A "Longitudinal Control Byte" is used. This is a simple, 8-bit XOR sum. *Its usage is strongly discouraged* because the receiver cannot detect a lost or an out-of-sequence packet, and the PPP frame checksum is a more reliable test of the packet's integrity.

2 CRC: A 16-bit cyclic redundancy checksum is used. Although this is a better test of a packet's integrity than the LCB, its use is still discouraged because the receiver still cannot use it to detect lost or out of sequence packets, and otherwise it becomes largely redundant with the frame checksum.

3 SEQ: An 8-bit sequence number is used (default). This is the preferred method of operation. If the number of histories is not 0, use of any other mode is strongly discouraged though another mode may be necessary for interoperability with certain non-RFC-compliant routers.

hdlc cable *cable type*
Set the HDLC cable type (that is connected to the interface) to one of the following.

- RS-232 DTE
- RS-232 DCE
- V35 DCE
- V35 DTE
- V36 DCE
- V36 DTE
- X21 DCE
- X21 DTE

Table 28-3 lists the cable types you can configure on the various adapters.

Table 28-3. Cable types for 2216 Interfaces	
Adapter Type	**Cable type**
8-port EIA 232	RS-232 DTE and RS-232 DCE
6-port V.35/V36	V.35 DCE, V.35 DTE, V.36 DCE, or V.36 DTE
8-port X.21	X.21 DCE and X.21 DTE

Example: `set hdlc cable rs-232 dce`

A DTE cable is used when you are attaching the router to some type of DCE device (for example, a modem or a DSU/CSU).

A DCE cable is used when the router is acting as the DCE and providing the clocking for direct attachment.

hdlc clocking *external* or *internal*
To connect to a modem or DSU, configure clocking as external. To connect directly to another DTE device, use a DCE cable and set the clocking to "internal" at one end and to "external" at the other. Configure the clock speed at the end using internal clocking. Use Table 28-4 on page 28-20 to determine the clock speeds you can set for the various adapters when internal clocking is used.

Example: `set hdlc clocking internal`

hdlc encoding *NRZ* or *NRZI*
> Sets the HDLC transmission encoding scheme for an interface. Encoding may be set for NRZ (non-return to zero) or NRZI (non-return to zero inverted). NRZ is the more widely used encoding scheme while NRZI is used in some IBM configurations. The default value is NRZ.

> **Example:**   `set hdlc encoding nrz`

hdlc idle *flag* or *mark*
> Sets the data link idle state to either Flag or Mark. The default value is Flag.

> The flag option provides continuous flags (7E hex) between frames.

> The mark option puts the line in a marking state (OFF, 1) between frames.

> **Example:**   `set hdlc idle flag`

hdlc speed *value*
> For internal clocking, this command specifies the speed of the transmit and receive clock lines. Refer to Table 28-4.

Table 28-4. Line Speeds When Internal Clocking is Used for 2216 Interfaces

Adapter Type	Speed Range
8-port EIA 232	9600 to 64 000 bps
6-port V.35/V.36	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps
8-port X.21	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps

> For external clocking, this command does not affect the hardware but it sets the speed some protocols, such as IPX, use to determine the routing parameters. In these cases, set the speed to match the actual line speed. Use Table 28-5 to determine the clock speeds you can set for the various adapters.

Table 28-5. Line Speeds When External Clocking is Used for 2216 Interfaces

Adapter Type	Speed Range
8-port EIA 232	2400 to 64 000 bps
6-port V.35/V.36	2400 to 2 048 000 bps
8-port X.21	2400 to 2 048 000 bps

> **Example:**   `set hdlc speed 56000`

hdlc transmit-delay *value*
> Sets the number of flags sent between frames. The purpose of this command is to slow the serial line so that it is compatible with older, slower serial devices at the other end.

> The range is 0 to 15. The default is 0.

> **Example:**   `set hdlc transmit-delay 15`

ipcp
> Sets all Internet Protocol Control Protocol options for that link.

> **Example:**   `set ipcp`

```
IP COMPRESSION [yes]:
Number of Slots:   [16]?
Send our IP address [yes]:
Request their IP address [yes]:
```

Note: The value immediately following a command option prompt is the current setting of that option. It is not always the default value illustrated in this chapter.

IPCP compression

> Selects whether or not the PPP handler will accept compressed IP data. PPP supports Van Jacobson (VJ) TCP/IP header compression as described in RFC 1144. You should enable this option when the point-to-point link is running at a low baud rate.

> Setting this value to yes enables the compression option. Setting this value to no disables the option. The default setting is no.

Slots

> Sets the number IP headers that are saved for referential purposes when determining the type of compression that is enabled. The range is 1 to 16. The default is 16.

Send our IP address

> Specifies whether or not to send the local IP address to the remote end of the link.   You should set this option to **Yes** if the other end of the link requires the IP address. In either case, PPP sends the local IP address if the other end of the link requests it.

Request their IP address

> Specifies whether or not to request the IP address from the remote end of the link. If PPP receives the remote IP address, it displays the address with the PPP monitoring statistics.

lcp *options* or *parameters*

> Sets the Link Control Protocol options and parameters for the PPP link.

> **Example:**   `set lcp options`

```
Maximum Receive Unit (bytes) [2048]?
Magic Number [yes]:
Peer-to-Local Async Control Character Map (RX ACCM) [A0000] ?
Protocol Field Compression (PFC) [no]?
Addr/Cntl Field Compression (ACFC) [no]?
```

Note: The value immediately following the command option prompt is the current setting of that option. It is not always the default value.

Maximum receive unit

> Sets the maximum packet size of the information field that are transferred in a single datagram. The range is 576 to 4089 bytes. The default is 2048.

Magic number

> Specifies whether or not the magic number option is enabled. The magic number provides a way of detecting looped back links in serial line configurations. When this option is enabled, the link uses the system clock as a random number generator. The random numbers that are generated are referred to as magic numbers.

> When the LCP receives a Configure Request with a magic number present (i.e., the magic number option is enabled), the received magic number is compared with the magic number in the last Configure-Request sent to the

peer. If the two magic numbers are different, the link is not considered looped back. If the two numbers are the same, the PPP handler attempts to bring the link down and up again to renegotiate magic numbers.

Setting this value to Yes enables the magic number option. Setting this value to No disables the option. The default setting is Yes.

Async Control Character Map
Indicates which characters that the peer must "escape" when transmitting packets to the router *on asynchronous lines*. This allows certain sensitive ASCII control characters, such as XON and XOFF, to be transmitted transparently over the link.

Specify a 32-bit bitmask in hexadecimal. If a bit in position 'N' of the mask is set, the corresponding ASCII character 'N' must be escaped (the LSB is bit number 0, corresponding to the ASCII NUL character).

The default value for this option is '0A0000', indicating that XON and XOFF (control-Q and control-S) need to be escaped. This is for the benefit of modems that use XON/XOFF to perform software handshaking. If this is not an issue, then it is recommended that you change the ACCM to zero (no characters escaped).

LCP is always willing to negotiate the ACCM, even on synchronous lines, and the **list lcp** command in the PPP monitoring process will display the negotiated value. However, synchronous lines employ a "bit-stuffing" mechanism rather than an "escaping" mechanism, so the ACCM is not normally meaningful on synchronous lines. It may be meaningful if the router is connected to a modem that performs sync-to-async conversion, in which case its value should reflect the requirements of the attached modem on the asynchronous side.

Addr/Cntl Field Compression (ACFC)
Specifies whether the peer can employ address and control field compression.

If the ACFC option is successfully negotiated by LCP, it means that the Address and Control field bytes which start off each packet may be omitted in the datagrams sent back and forth on the link. These bytes are always 0xFF 03, so there is no real information provided by them, and enabling ACFC means that the datagrams that are transmitted will be two bytes shorter.

To be precise, if you enable ACFC, you are indicating a receive-side capability. If you enable ACFC and LCP successfully negotiates it, the other end can employ ACFC in the packets it transmits to the local end (most PPP options work like this). The local end will only transmit packets *without* the address and control fields if the other end also indicates its ability to handle such packets.

Enabling ACFC does not obligate the other end to send packets without the address and control fields, even if it accepts the option. Enabling ACFC merely tells the peer that it optionally *may* use ACFC, and the router will be able to handle the incoming packets. If the peer indicates that it can handle ACFC, then the router always performs ACFC on the packets it transmits regardless of whether ACFC is enabled locally.

LCP packets always are sent with address and control fields present. This guarantees that LCP packets will be recognized even if there is a loss of link synchronization.

Protocol Field Compression (PFC)
Specifies whether the peer is to employ protocol field compression.

When you specify "yes," if the PFC option is negotiated successfully by LCP, the leading zero byte may be omitted from the "Protocol" field for those protocol values in the range '0x0000'–'0x00FF', for a one byte savings in the packets being transmitted. This range includes the majority of layer-3 protocol datagrams.

PPP protocol values are all assigned such that the upper byte of the protocol is an even value and the lower byte is an odd value (a limited use of the more generalized mechanism described by the ISO 3309 extension mechanism for address fields). Thus, the receiver can readily detect when the leading byte of a protocol value has been omitted (the first byte of the protocol field is odd rather than even), so there is no ambiguity interpreting frames in the presence of PFC.

PFC, like ACFC, is a receive side capability and the previous description of ACFC applies to PFC. The protocol value for LCP is '0xC021' which does not fall in the range '0x0000'–'0x00FF'; therefore, *LCP packets are never candidates for PFC*.

Example: `set lcp parameters`

```
Config tries  [20]?
NAK tries  [10]?
Terminate tries  [10]?
Retry timer (mSec)  [3000]?
```

Note: The value immediately following the command option prompt is the current setting of that option. It is not always the default value illustrated in this chapter.

Retry timer
: Sets the amount of time in milliseconds that elapses before LCP's transmission of configure-request (to open the link) and terminate-request (to close the link) packets is timed out. Expiration of this timer causes a timeout and the halting of configure-request and terminate-request packet transmission. The range is 200 to 30000 milliseconds. The default setting is 3000 milliseconds.

Config tries
: Sets the number of times that LCP sends configure-request packets to a peer station to establish the opening of a PPP link. The default value is 20. The range is 1 to 100.

: The retry timer starts after the first configure-request packet is transmitted. This is done to guard against packet loss.

NAK tries
: Sets the number of times that LCP sends configure-nak (nak = not acknowledged) packets to a peer station while attempting to open a PPP link. The default value is 10. The range is 1 to 100.

LCP sends configure-nak packets upon receiving configure-request packets with some unacceptable configuration options. These packets are sent to refuse the offered configuration options and to suggest modified, acceptable values.

Terminate tries   Sets the number of times that LCP sends terminate-request packets to a peer station to close a PPP link. The default value is 10. The range is 1 to 100.

The retry timer starts after the first terminate-request packet is transmitted. This is done to guard against packet loss.

name *router name*

Sets the name that the router uses when responding to authentication requests from another router.

Note: While the "case" you use for names and passwords sent to the peer on the link are preserved for this product, interoperability with other vendor products is easier if all names and passwords are entered in *lower* case.

Example: **set name routerid**

```
PPP Config> set name router23
enter password: cd3456
enter password again: cd3456
remote name set successfully
```

ncp parameters

Sets the parameters for all NCPs.

Example: **set ncp parameters**

```
Config tries  [20]
NAK tries  [10]?
Terminate tries  [10]?
Retry timer (mSec)  [3000]?
```

Config tries   Sets the number of configure-request packets sent by NCP to a peer station to attempt to open a PPP link. The range is 1 to 100. The default is 20.

This action indicates the desire to open an NCP connection with a specified set of configuration options. The retry timer starts after a configure-request packet is transmitted. This is done to guard against packet loss.

NAK tries   Sets the number of configure-nak (nak = not acknowledged) packets that NCP sends to a peer station while attempting to open a PPP link. The range is 1 to 100. The default value is 10.

Upon receiving configure-request packets with some unacceptable configuration options, NCP sends configure-nak packets. These packets are sent to refuse the offered configuration options and to suggest modified, acceptable values.

Terminate tries	Sets the number of terminate-request packets sent by NCP to a peer station to close a PPP link. The range is 1 to 100. The default value is 10.
	This action indicates the desire to close an NCP connection. The retry timer is started after a terminate-request packet is transmitted. This is done to guard against packet loss.
Retry timer	Sets the amount of time, in milliseconds, that elapses before NCP's transmission of configure-request (to open the link) and terminate-request (to close the link) packets is timed out. Expiration of this timer causes a timeout and the halting of configure-request and terminate-request packet transmission. The range is 200 to 30000 milliseconds. The default is 3000 milliseconds.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: `exit`

Chapter 29. Monitoring Point-to-Point Protocol Interfaces

This chapter describes how to monitor specific Point-to-Point Protocol interfaces in the router. Sections in this chapter include:

- "Accessing the Interface Console Process"
- "Point-to-Point Console Commands"
- "Point-to-Point Protocol Interfaces and the GWCON Interface Command" on page 29-16

Accessing the Interface Console Process

To access the PPP interface console process, do the following:

1. Enter **interface** at the + prompt to display a list of configured interfaces.

2. Enter **network** followed by the number of the PPP interface.

   ```
   + network 2
   PPP>
   ```

Point-to-Point Console Commands

This section summarizes and then explains the Point-to-Point console commands. Enter the commands at the PPP> prompt. Table 29-1 shows the commands.

Note: The options available for these commands depend on what what protocols are available in the router software. For example, when the router software (image) does not contain APPN support, the **list isrcp**, **list isr**, **list hprcp**, **list hpr**, and **llc** commands are not available.

Table 29-1. Point-to-Point Console Command Summary	
Command	**Function**
? (Help)	Displays all the Point-to-Point commands or lists subcommand options for specific commands (if available).
Clear	Clears all statistics from point-to-point interfaces.
List	Displays information and counters related to the point-to-point interface and PPP parameters and options.
LLC	Displays the LLC console monitoring prompt.
Exit	Exits the Point-to-Point console process.

? (Help)

Use the **? (Help)** command to obtain a list of the commands available from that prompt level. You can also enter this command after specific command names to obtain a listing of the command options available for that command.

Syntax: ?

Example: **?**

```
CLEAR
LIST
LLC
EXIT
```

Example: `list ?`

Clear

Use the **clear** command to clear all statistics from point-to-point interfaces.

Syntax: c̲lear

Example: `clear`

List

Use the **list** command to display information and counters related to the
point-to-point interface and PPP parameters and options.

Syntax: l̲ist   a̲ll
                    c̲ontrol
                    e̲rrors
                    i̲nterface
                    l̲cp   - PPP link CP
                    p̲ap   - PAP Authentication CP
                    c̲hap   - CHAP Authentication CP
                    c̲cp   - PPP Compression CP
                    c̲dp   - PPP compression
                    b̲cp   - Bridging (ASRT) CP
                    b̲rg   - Bridging (ASRT)
                    s̲tp   - Spanning Tree Protocol
                    i̲pcp   - Internet Protocol CP
                    i̲p   - Internet Protocol
                    i̲pxcp   - Novell IPX CP
                    i̲px   - Novell IPX
                    a̲tcp   - Appletalk (Phase 2) CP
                    a̲p2   - Appletalk (Phase 2)
                    d̲ncp   - DECnet IV CP
                    d̲n   - DECnet IV
                    o̲sicp   - ISO's OSI CP
                    o̲si   - ISO's OSI
                    b̲vcp   - Banyan VINES CP
                    v̲ines   - Banyan VINES
                    i̲srcp   - APPN ISR CP
                    i̲sr   - APPN ISR
                    h̲prcp   - APPN HPR CP
                    h̲pr   - APPN HPR

all

Lists all information and counters related to the point-to-point interface and
PPP options and parameters. The output displayed for this command is a
combination of the **list control lcp** and **list lcp** commands, the **list error
command**, the **list interface** command, the command for listing each of the
authentication protocols, the commands for listing data compression (**list
control ccp**, **list ccp**, **list cdp**), and the commands for listing each network
protocol (for IP, for example, the **list control ipcp**, **list ipcp**, and **list ip**
commands).

Note: If a network control protocol is not available on an interface, a message
is displayed indicating that no protocol or statistics information is available for
that network control protocol's list commands.

Example: `list all`

control

Lists negotiated options or other state information for a control protocol.

    ccp
    lcp
    bcp
    ipcp
    ipxcp
    atcp
    dncp
    osicp
    bvcp
    isrcp
    hprcp

Example: `list control ccp`

```
CCP State:              Open
Previous State:         Ack Sent
Time Since Change:      264 hours, 56 minutes and 58 seconds

Compressor:   STAC-LZS histories 1, check_mode SEQ
Decompressor: STAC-LZS histories 1, check_mode SEQ

Max size of compression dictionary:   12494.
Max size of decompression dictionary: 4424.
```

CCP state

current state of the point-to-point link.

Previous State

State of the point-to-point link before the state displayed in the current state field.

Compressor

Shows which compressor was negotiated and the options it is using.

Decompressor

Shows which decompressor was negotiated and the options it is using.

Max size of compression dictionary

The size of the data space allocated for the compression "context" or "history."

Max size of decompression dictionary

The size of the data space allocated for the decompression "context" or "history."

Example: `list control lcp`

```
Version:                1
        Link phase:             Establishing connection (LCP)
        LCP State:              Listen
        Previous State:         Req Sent
        Time Since Change:      1 minute and 57 seconds
        Remote Username:        - No Authentication -
        Last Identification Rx'd
        Time Connected:         - No Connection -

        LCP Option              Local                   Remote
        ----------              -----                   ------
        Max Receive Unit:       2048                    1500
        Async Char Mask:        FFFFFFFF                FFFFFFFF
        Authentication:         None                    None
        Magic Number:           7A8CBFD7                None
        Protocol Field Comp:    No                      No
```

```
Addr/Cntl Field Comp:       No                    No
32-Bit Checksum:            No                    No
```

Version
> Displays the current version of the Point-to-Point Protocol.

LCP State
> Displays the current state of the point-to-point link. These states include the following:

> OPEN - Indicates that a connection has been made and data may be sent. The retry timer does not run in this state.

> CLOSED - Indicates that the link is down and there is no attempt being made to open it. In this state, all connection requests from peers are rejected.

> LISTEN - Indicates that the link is down and there is no attempt being made to open it. In contrast to the CLOSED state, however, all connection requests from peers are accepted.

> REQUEST-SENT - Indicates that an active attempt is being made to open the link. A Configure-request packet has been sent but a Configure-Ack has not yet been received nor has one been sent. The retry timer is running at this time.

> ACK-RECEIVED - Indicates that a Configure-request packet has been sent and a Configure-Ack packet has been received. The retry timer is still running since a Configure-Ack packet has not been transmitted.

> ACK-SENT - Indicates that a Configure-Ack packet and a Configure-request packet have been sent but a Configure-Ack packet has not been received. The retry timer always runs in this state.

> CLOSING - Indicates that an attempt is being made to close the connection. A Terminate-request packet has been sent but a Terminate-Ack packet has not been received. The retry timer is running in this state.

Previous State
> Displays the state of the point-to-point link prior to the state displayed in the Current state field. These states are the same as those described in the Current state field.

Time since change
> Displays the amount of time that has elapsed since the last link state change.

Remote Username
> When authentication is required on the link, this field shows the name that the peer supplied.

Last Identification Rx'd
> An optional packet type that is defined for LCP as an "Identification" packet. The contents of this packet are undefined but is normally expected to be a human readable string provided by the peer to provide some identifying information such as a name, manufacturer, model number, or other information the manufacturer wishes to provide. If the router receives such a packet, the contents of the last such packet received is displayed here.

Time Connected
>   Indicates how long the peer has been connected on this link.

LCP Option
>   These fields indicate the values of options that have been negotiated with the peer when LCP is in the Open state. When LCP is not open, these values represent initial defaults or configured values that will be used in subsequent LCP negotiations.

Max Receive Unit
>   Indicates the maximum length for the packet size that the local and remote ends can transmit. This is the maximum length of the payload portion of a PPP packet and it *does not include* PPP header and trailer bytes.
>
>   When LCP is in an Open state, the values indicate the lengths that have been negotiated with the peer. The router does not support differing MRU lengths for the peer and local end, so these values will be the same.

Async Character Mask
>   This indicates the asynchronous control character mask that has been negotiated. The router accepts ACCM negotiation even on synchronous lines, although this does not affect the actual packet data sent. See Chapter 28, "Using and Configuring Point-to-Point Protocol Interfaces" on page 28-1 for more information about the ACCM.

Authentication
>   Indicates which authentication protocol, if any, each end of the link requires. Multiple protocols may be available at each end; this value indicates which protocol the units *agreed* to use.

Magic number
>   Displays the current magic number being used for both the local and remote ends of the link for loopback detection.

Protocol compression
>   Indicates whether PFC has been negotiated.

Address/Control compression
>   Indicates whether ACFC has been negotiated.

32-bit checksum
>   Not currently supported. PPP will reject this option if it is received.

Example: `list control bcp`

```
BCP State:          Closed
Previous State:      Closed
Time Since Change:   5 hours, 25 minutes and 3 seconds

BCP Option          Local                   Remote
Tinygram Compression DISABLED               DISABLED
Source-route Info:
Remote side does not support source-route bridging
```

The BCP State fields are the same as those described under the **list control lcp** command.

Tinygram Compression
>   Displays whether or not Tinygram Compression is enabled or disabled on the local and remote ends of the link.

Source-route Info
> Displays whether or not source route bridging is enabled for the local and remote ports that correspond to this interface.

Example: `list control ipcp`

```
IPCP State:          Listen
Previous State:      Closed
Time Since Change:   1 hour, 57 minutes and 52 seconds

IPCP Option          Local                    Remote
-----------          -----                    ------
IP Address           0.0.0.0                  None
Compression Slots    None                     None
```

The IPCP State fields are the same as those described under the **list control lcp** command.

Compression Slots
> Indicates the number of IP headers saved for referential purposes when determining the type of compression that is enabled.

IP Address:
> Indicates if this device's IP address is sent and if the other side's IP address is requested or not.

Example: `list control ipxcp`

```
IPXCP State:         Closed
Previous State:      Closed
Time Since Change:   2 hours, 9 minutes and 9 seconds
```

The IPXCP state fields are the same as those described under the **list control lcp** command.

Example: `list control atcp`

```
ATCP State:          Closed
Previous State:      Closed
Time Since Change:   6 hours, 27 minutes and 7 seconds

AppleTalk Address Info:
Common network number = 12
Local node ID = 49
Remote node ID = 76
```

The ATCP State fields are the same as those described under the **list control lcp** command.

Common Network Number
> Network number of the two ends of the point-to-point link. (You must statically configure both ends of the link to have the same network number.)

Local Node ID
> Unique node number of the local end of the link.

Remote Node ID
> Unique node number of the remote end of the link.

Example: `list control dncp`

```
DNCP State:          Closed
Previous State:      Closed
Time Since Change:   2 hours, 2 minutes and 58 seconds
```

The DNCP state fields are the same as those described under the **list control lcp** command.

Example: `list control osicp`

```
OSICP State:        Closed
Previous State:     Closed
Time Since Change:  6 hours, 28 minutes and 32 seconds
```

The OSICP State fields are the same as those described under the **list control lcp** command.

Example: `list control bvcp`

```
BVCP State:         Open
Previous State:     Ack Sent
Time Since Change:  403 hours, 49 minutes and 2 seconds
```

The BVCP State fields are the same as those described under the **list control lcp** command.

Note: The command word **bvcp** and the acronym BVCP stand for the Banyan VINES Control Protocol (BVCP).

Example: `list control isrcp`

```
APPNISRCP State:    Open
Previous State:     Ack Rcvd
Time Since Change:  1 hour, 48 minutes and 5 seconds
```

The APPN ISR control protocol (ISRCP) state fields are the same as those described under the list control lcp command.

Example: `list control hprcp`

```
APPN HPRCP State:   Open
Previous State:     Ack Rcvd
Time Since Change:  1 hour, 48 minutes and 10 seconds
```

The APPN HPR control protocol (HPRCP) state fields are the same as those described under the list control lcp command

error

Lists information related to all error conditions tracked by the PPP software.

Example: `list error`

```
Error Type              Count        Last One
----------              -----        --------
Bad Address:              0             0
Bad Control:              0             0
Unknown Protocol:         0             0
Invalid Protocol:         0             0
Config Timeouts:          0             0
Terminate Timeouts:       0             0
```

Bad address
   Indicates the total number of bad addresses encountered over the point-to-point link. "Bad addresses" refers to the HDLC framing byte at the start of the packet.

Bad control
   Indicates the total number of bad control packets encountered over the point-to-point link. "Bad control" refers to the 0x03 prefix on HDLC encapsulated PPP packets ("UI" value that follows the 0xFF).

Unknown protocol
   Indicates the total number of unknown protocol packets encountered by the current link.

Invalid protocol
Indicates the total number of invalid protocol packets encountered by the current link.

Config timeouts
Indicates the total number of configuration timeouts experienced by the link.

Terminate timeouts
Indicates the total number of link termination timeouts experienced by the link.

interface
Lists PPP interface statistics.

Example: `list interface`

```
Interface Statistic        In              Out
------------------         --              ---
Packets:                   0               0
Octets:                    0               0
```

Packets
Indicates the number of packets received and transmitted on this interface.

Octets
Indicates the number of octets received and transmitted on this interface.

lcp
Lists statistics for the Link Control Protocol.

Example: `list lcp`

```
LCP STATISTIC       IN       OUT
-------------       --       ---
PACKETS:            42       42
OCTETS:             1260     1260
CFG REQ:            0        0
CFG ACK:            0        0
CFG NAK:            0        0
CFG REJ:            0        0
TERM REQ            0        0
TERM ACK            0        0
ECHO REQ:           21       21
ECHO RESP:          21       21
DISC REQ:           0        0
CODE REJ:           0        0
```

Packets
Indicates the total number of LCP packets transmitted (out) and received (in) over the current point-to-point interface.

Octets
For LCP frames, indicates the total number of bytes in octets transmitted and received over the current point-to-point interface.

CFG REQ
Indicates the total number of configure-request LCP packets transmitted and received over the current point-to-point interface.

CFG ACK
Indicates the total number of configure-ack (acknowledged) LCP packets transmitted and received over the current point-to-point interface.

CFG NAK
Indicates the total number of configure-nak (not acknowledged) LCP packets transmitted and received over the current point-to-point interface.

CFG REJ
> Indicates the total number of configure-reject LCP packets transmitted and received over the current point-to-point interface.

TERM REQ
> Total number of terminal request LCP packets transmitted and received over the current point-to-point interface.

TERM ACK
> Total number of terminal ack LCP packets transmitted and received over the current point-to-point interface.

ECHO REQ
> Indicates the total number of echo-request LCP packets transmitted and received over the current point-to-point interface.

ECHO RESP
> Indicates the total number of echo-response LCP packets transmitted and received over the current point-to-point interface.

DISC REQ
> Indicates the total number of discard-request LCP packets transmitted and received over the current point-to-point interface.

CODE REJ
> Indicates the total number of code-reject LCP packets transmitted and received over the current point-to-point interface.

pap
> Lists statistics for the Password Authentication Protocol.

> **Example: `list pap`**

```
PAP Statistics          In                      Out
--------------          --                      ---
Packets:                0                       0
Octets:                 0                       0
Requests:               0                       0
Acks:                   0                       0
Naks:                   0                       0
```

Packets
> The total number of PAP packets sent or received.

Octets
> The number of bytes of data that were sent or received in those packets.

Requests
> The number of PAP "Request" packets sent or received. These are the packets which contain the PAP name/password pairs.

Acks
> The number of Acks (success replies) sent or received for the PAP requests (for example, if the peer sends a valid Request packet, the router replies with an Ack).

Naks
> The number of Naks sent or received for the PAP requests (for example, if the peer sends an invalid Request packet, the router replies with a Nak).

chap
>   Lists statistics for the Challenge-Handshake Authentication Protocol.
>
>   **Example:** `list chap`
>
>    ```
>    CHAP Statistics         In                   Out
>    ---------------         --                   ---
>    Packets:                0                    0
>    Octets:                 0                    0
>    Challenges:             0                    0
>    Responses:              0                    0
>    Successes:              0                    0
>    Failures:               0                    0
>    ```

Packets
>   The total number of CHAP packets sent or received.

Octets
>   The number of bytes of data that were sent or received in the packets.

Challenges
>   The number of CHAP "Challenge" packets sent or received. A CHAP
>   Challenge packet includes a randomly generated encryption key and is a
>   demand on the peer to generate a suitable response based on that key
>   and on stored password information.

Responses
>   The number of CHAP "Response" packets sent or received. A Response
>   packet contains a peer's answer to a "Challenge" request.

Successes/Failures
>   The number of Success or Failure packets sent or received. A unit sends
>   out a Challenge packet and waits for the peer's Response reply. It then
>   examines the Response packet and sends a Success or Failure packet to
>   indicate whether the Response was valid.
>
>   These counters reflect the number of Success or Failure packets sent. A
>   peer gets several tries to respond successfully before authentication is
>   considered to have failed.

ccp
>   Lists statistics for compression control protocol.
>
>   **Example:** `list ccp`
>
>    ```
>    CCP   Statistic         In                   Out
>    ---------------         --                   ---
>    Packets:                24                   25
>    Octets:                 174                  177
>    Reset Reqs              0                    0
>    Reset Acks:             0                    0
>    Prot Rejects:           0                    0
>    ```

Packets
>   Indicates the number of packets received and transmitted on this interface.

Octets
>   Indicates the number of octets received and transmitted on this

Reset Reqs
>   The number of CCP dictionary "Reset Requests" that were transmitted or
>   received.

Reset Acks

> The number of CCP dictionary "Reset Acknowledgements" that were transmitted or received.
>
> Reset Request and Reset Acknowledgement packets are control packets passed between the CCP entities at each end, used to maintain synchronization of the data dictionaries at each end of the link.

Prot Rejects

> Indicates the number of protocol rejects of CCP packets sent by the peer (reception of a protocol reject would signify that the peer does not support CCP).

cdp

Displays statistics for compressed data packets.

```
Compression Statistic     In                   Out
---------------------     --                   ---
Packets:                  31035                46550
Octets:                   1614885              2421137
Compressed Octets:        931416               1521039
Incompressible Packets:   0                    0
Discarded Packets:        0                    0
Copied Packets:           1                    0
Prot Rejects:             0                    -

Compressor (transmit) statistics:
    Recent compression ratio: 1.7:1
Decompressor (receive) statistics:
    Recent compression ratio: 1.7:1
```

bcp

Lists statistics for the Bridging control protocol. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list bcp

```
BCP Statistic         In                   Out
--------------        --                   ---
Packets:              0                    0
Octets:               0                    0
Prot Rejects:         0                    -
```

brg

Lists statistics on the bridge packets received and transmitted over the PPP interface. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list brg

```
BRG Statistic         In                   Out
--------------        --                   ---
Packets:              0                    0
Octets:               0                    0
Prot Rejects:         0                    -
```

stp

Lists statistics for the spanning tree protocol. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list stp

```
Spanning Tree Statistic    In           Out
-----------------------    --           ---
Packets:                   0            0
Octets:                    0            0
```

ipcp

Lists Internet Protocol Control Protocol statistics for the point-to-point interface. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list ipcp

```
IPCP STATISTIC    IN        OUT
--------------    --        ---
PACKETS:          0         0
OCTETS:           0         0
PROT REJECTS:     0
```

ip

Lists all information related to IP packets over the point-to-point link.

Example: list ip

```
IP Statistic      In        Out
------------      --        ---
Packets:          349       351
Octets:           128488    129412
Prot Rejects:     0         -
```

Packets

Indicates the total number of IP packets transmitted (out) and received (in) over the current point-to-point interface.

Octets

Indicates the total number of octets of data bytes transmitted and received over the current IP connection.

Prot Rejects

Indicates the total number of protocol reject packets transmitted and received over the current point-to-point interface.

ipxcp

Lists statistics for the IPX control protocol. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list ipxcp

```
IPXCP Statistic     In             Out
--------------      --             ---
Packets:            0              0
Octets:             0              0
Prot Rejects:       0              -
```

ipx

Lists IPX statistics for the point-to-point interface. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list ipx

```
IPX Statistic     In        Out
------------      --        ---
Packets:          0         0
Octets:           0         0
Prot Rejects:     0         -
```

atcp

Lists statistics for the AppleTalk control protocol. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list atcp

```
ATCP Statistic       In              Out
---------------      --              ---
Packets:             0               0
Octets:              0               0
Prot Rejects:        0               -
```

ap2

Lists AppleTalk Phase 2 statistics for the point-to-point interface. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list ap2

```
AP2 Statistic    In        Out
------------     --        ---
Packets:         349       351
Octets:          128488    129412
Prot Rejects:    0
```

dncp

Lists statistics on the DECnet control protocol packets. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list dncp

```
DNCP Statistic       In              Out
--------------       --              ---
Packets:             0               0
Octets:              0               0
Prot Rejects:        0               -
```

dn

Lists statistics on the DECnet packets received and transmitted over the PPP interface. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list dn

```
DN Statistic     In        Out
------------     --        ---
Packets:         0         0
Octets:          0         0
Prot Rejects:    0         -
```

osicp

Lists statistics for the OSI control protocol. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list osicp

```
OSICP Statistic      In              Out
---------------      --              ---
Packets:             0               0
Octets:              0               0
Prot Rejects:        0               -
```

osi

Lists statistics on the OSI packets received and transmitted over the PPP interface. These fields are the same as those described under the **list ip** command. (See page 29-12.)

Example: list osi

```
OSI Statistic    In        Out
-------------    --        ---
Packets:         0         0
Octets:          0         0
Prot Rejects:    0         -
```

bvcp

Lists statistics on the Banyan VINES control protocol. These fields are the
same as those described under the **list ip** command. (See page 29-12.)

Example: list bvcp

```
BVCP Statistic        In                 Out
--------------        --                 ---
Packets:              0                  0
Octets:               0                  0
Prot Rejects:         0                  -
```

vines

Lists statistics for the Banyan VINES packets received and transmitted over the
PPP interface. These fields are the same as those described under the **list ip**
command. (See page 29-12.)

Example: list vines

```
Vines Statistic        In                 Out
---------------        --                 ---
Packets:               10                 13
Octets:                320                340
Prot Rejects:          0                  -
```

isrcp

Lists statistics for APPN ISR Control Protocol packets. These fields are the
same as those described under the **list ip** command. (See page 29-12.)

Example: list isrcp

```
APPNISRCP Statistic    In                 Out
---------------        --                 ---
Packets:               3                  3
Octets:                12                 12
Prot Rejects:          0                  -
```

isr

Lists statistics on the APPN ISR packets received and transmitted over the
PPP interface. These fields are the same as those described under the **list ip**
command. (See page 29-12.)

Example: list isr

```
APPNISR Statistic      In                 Out
-------------          --                 ---
Packets:               220                219
Octets:                1266               1157
Prot Rejects:          0                  -
```

hprcp

Lists statistics for APPN HPR Control Protocol packets. These fields are the
same as those described under the **list ip** command. (See page 29-12.)

Example: list hprcp

```
APPNHPRCP Statistic    In                 Out
---------------        --                 ---
Packets:               3                  3
Octets:                12                 12
Prot Rejects:          0                  -
```

hpr

Lists statistics on the APPN HPR packets received and transmitted over the
PPP interface. These fields are the same as those described under the **list ip**
command. (See page 29-12.)

Example: list hpr

```
APPNHPR Statistic      In                Out
-------------          --                ---
Packets:               780               715
Octets:                131907            69685
Prot Rejects:          0                 -
```

LLC

Use the **LLC** command to access the LLC monitoring prompt. LLC commands are entered at this new prompt. See "LLC Monitoring Commands" on page 20-1 for an explanation of each of these commands.

Note: This command is available only when APPN is included in the software load.

Syntax: l̲l̲c̲

Example: llc

```
LLC user monitoring
LLC>
```

Exit

Use the **exit** command to return to the GWCON prompt level.

Syntax: e̲xit

Example: exit

Point-to-Point Protocol Interfaces and the GWCON Interface Command

While Point-to-Point interfaces have their own console processes for monitoring purposes, the device also displays complete statistics for installed network interfaces when you use the **interface** command from the GWCON environment. (For more information on the **interface** command, refer to Chapter 6, "The GWCON (Monitoring) Process and Commands.")

The statistics in this section display when you run the **interface** command from the GWCON environment for the following interfaces used in point-to-point configurations:

Example: `interface 12`

```
                                             Self-Test Self-Test Maintenance
Nt Nt'  Interface Slot-Port              Passed    Failed    Failed
12 12   PPP/0     Slot: 8  Port: 2          2         1         0

 Point to Point MAC/data-link on V.35/V.36 interface

 Adapter cable:              V.35 DTE

 V.24 circuit: 105 106 107 108 109
 Nicknames:    RTS CTS DSR DTR DCD
 PUB 41450:    CA  CB  CC  CD  CF
 State:        ON  ON  ON  ON  ON

 Line speed:             64.000 Kbps
 Last port reset:        1 hour, 20 minutes, 42 seconds ago

 Input frame errors:
  CRC error                      0    alignment (byte length)         0
  missed frame                 182    too long (>  2062 bytes)        0
  aborted frame                  0    DMA/FIFO overrun                0
 Output frame counters:
  DMA/FIFO underrun errors       0    Output aborts sent              0
```

Nt
> Interface number as assigned by software during initial configuration.

Nt'
> Base interface number as assigned by software during initial configuration.

> **Note:** For dial circuit interfaces, Nt' is different from Nt. For dial circuit interfaces, Nt' indicates the base interface (ISDN or V.25bis) that the dial circuit uses.

Interface No
> Type of interface and its instance number. The Point-to-Point interface type is PPP.

Slot
> The slot number of the interface over which PPP is running.

Port
> The port number of the interface that is running PPP.

Self-Test: Passed
> Total number of times the point-to-point interface passed its self-test.

Self-Test: Failed
> Total number of times the point-to-point interface failed its self-test.

Maintenance: Failed
>   Total number of maintenance failures.

Adapter cable
>   Type of adapter cable that has been configured; for example, V.35 DTE.

V.24 circuit
>   Circuits being used on the V.24. Note: The symbol - - - in console output
>   indicates that the value or state is unknown.

Nicknames
>   Control signals Note: The symbol - - - in console output indicates that the
>   value or state is unknown.

PUB 41450
>   Pin assignments Note: The symbol - - - in console output indicates that the
>   value or state is unknown.

State
>   State of the V.24 circuits (on or off). Note: The symbol - - - in console
>   output indicates that the value or state is unknown.

Line speed
>   Configured line speed or default value assumed (if line speed is configured as
>   0).

Last port reset
>   Length of time since the port was reset.

CRC error
>   The number of packets received that contained checksum errors and as a
>   result were discarded.

Alignment (byte length)
>   The number of packets received that were not an even multiple of 8 bits in
>   length and as a result were discarded.

Too long (> 2048 bytes)
>   The number of packets that were greater than the configured frame size, and
>   as a result were discarded.

Aborted frame
>   The number of packets received that were aborted by the sender or a line
>   error.

DMA/FIFO overrun
>   The number of times the serial interface could not send data fast enough to the
>   system packet buffer memory to receive them from the network.

Missed frame
>   When a frame arrives at the device and there is no buffer available, the
>   hardware drops the frame and increments the missed frame counter.

L & F bits not set
>   On serial interfaces, the hardware sets input-descriptor information for arriving
>   frames. If the buffer can accept the complete frame upon arrival, the hardware
>   sets both the last and first bits of the frame, indicating that the buffer accepted
>   the complete frame. If either of the bits is not set, the packet is dropped, the `L
>   & F bits not set` counter is incremented, and the buffer is cleared for reuse.
>
>   **Note:** It is unlikely that the `L & F bits not set` counter will be affected by
>   traffic.

Output Frame Counters:

DMA/FIFO underrun errors
> The number of times the serial interface could not retrieve data fast enough
> from the system packet buffer memory to transmit them onto the network.

Output aborts sent
> The number of transmissions that were aborted as requested by upper-level
> software.

Statistics similar to the following are displayed for PPP dial circuits when you
execute the **interface** command from the GWCON environment:

```
+interface 29
                                        Self-Test  Self-Test  Maintenance
Nt Nt'  Interface                       Passed     Failed     Failed
29 10   PPP/20                             2          1          0

  Point to Point MAC/data-link on V.25bis Dial Circuit interface
```

Chapter 30. Data Compression for PPP Interface

This chapter discusses data compression on the router at PPP interfaces. It includes these sections:

- "Data Compression Concepts"
- "Data Compression on the Router" on page 30-2
- "Configuring and Monitoring Data Compression" on page 30-4

Data Compression Concepts

Data compression lets the router software pack more user data into the packets moving through router interfaces. If the rate of packet transfer stays constant, the data transfer rate (throughput) for the router increases proportionally.

In general, data compression is much more effective when used in conjunction with lower speed WAN interfaces, for example, less than 64 Kbps. As you increase the WAN data rate, the effective improvement in the overall bandwidth utilization diminishes. If the link data rate is sufficiently high, enabling compression *decreases* router performance and throughput.

Character Sequence Substitution

The data compression software works by replacing frequently occurring character sequences with single characters that represent the sequences. The sequences can be words, blank spaces, numbers, or any other string in the data stream. For example, the character sequences that make up the words "the" and "computer" may appear throughout an ASCII text file in a data stream. The data compression software substitutes a single character for each of the words. Further, if the phrase "the computer" appears, a different index character replaces the phrase. This substitution scheme in effect compresses the data.

Dictionaries

Data compression *dictionaries* maintain an association between each repeatedly occurring sequence and its substitution character. The dictionaries must be the same at the transmitting and receiving routers to ensure accurate compression and decompression. Compression protocol messages implement the dictionary coordination; and the router must monitor the communications link between the routers closely to maintain the coordination.

Compression Protocols and Algorithms

The router software implements the following data compression protocols and algorithms. The data compression software runs in the router's network layer.

- Compression Control Protocol (CCP).

  The router uses this protocol to negotiate the choice of a compression algorithm (see below) with the other router on the link.

  CCP is defined in RFC 1962.

- Compression algorithms.

Compression algorithms are software that carry out the actual compression of outgoing data and decompression of incoming data. The algorithms use an independent compression dictionary for each direction on the WAN link.

An important consideration in choosing a compression algorithm is its *compression ratio*: the size of the original data stream relative to the compressed data stream. The greater the ratio the more effective the compression. Compression algorithms typically have compression ratios of 2:1 and 3:1. The higher the compression ratio that you configure, the greater the burden of the CPU. Such a burden could lower the packets per second.

Note: The router supports only one algorithm, Stac Electronics' LZS-221 algorithm as defined in RFC 1974.

The compression ratio depends heavily on the data to be transmitted. ASCII English text is highly compressible (with a large compression ratio). Executable binary data is generally moderately compressible. Data that is precompressed by an application (for example, compressed image data streams like MPEG or JPEG) is incompressible.

- Compression algorithm interfaces.

  These are modules that interface between the PPP protocol software and a particular compression algorithm.

Data Compression on the Router

This section discusses how data compression is implemented on the router.

Protocols and Interfaces that Support Data Compression

The router supports data compression on PPP serial and dial circuit interfaces. The router implements data compression on a per-interface basis for each of the router's PPP interfaces. You enter data compression commands at the PPP> interface along with the other commands to configure and monitor PPP.

In the network layer, the router implements the CCP protocol as a Network Control Protocol (NCP). See Chapter 28, "Using and Configuring Point-to-Point Protocol Interfaces" on page 28-1. It also implements a compression algorithm for the STAC data compression protocol.

The Data Compression Process

The following sections and Figure 30-1 on page 30-3 describe the data compression process on the router.

PPP data compression is supported on PPP serial line interfaces as well as PPP dial circuits that are mapped to V.25bis and ISDN interfaces.

Negotiating a Compression Algorithm

When the router brings up a PPP link on the WAN, it initiates an exchange of CCP packets as part of the link establishment. The router negotiates with its peer router the use of data compression and compression algorithm parameters (for example, the check mode used by the Stac algorithm).
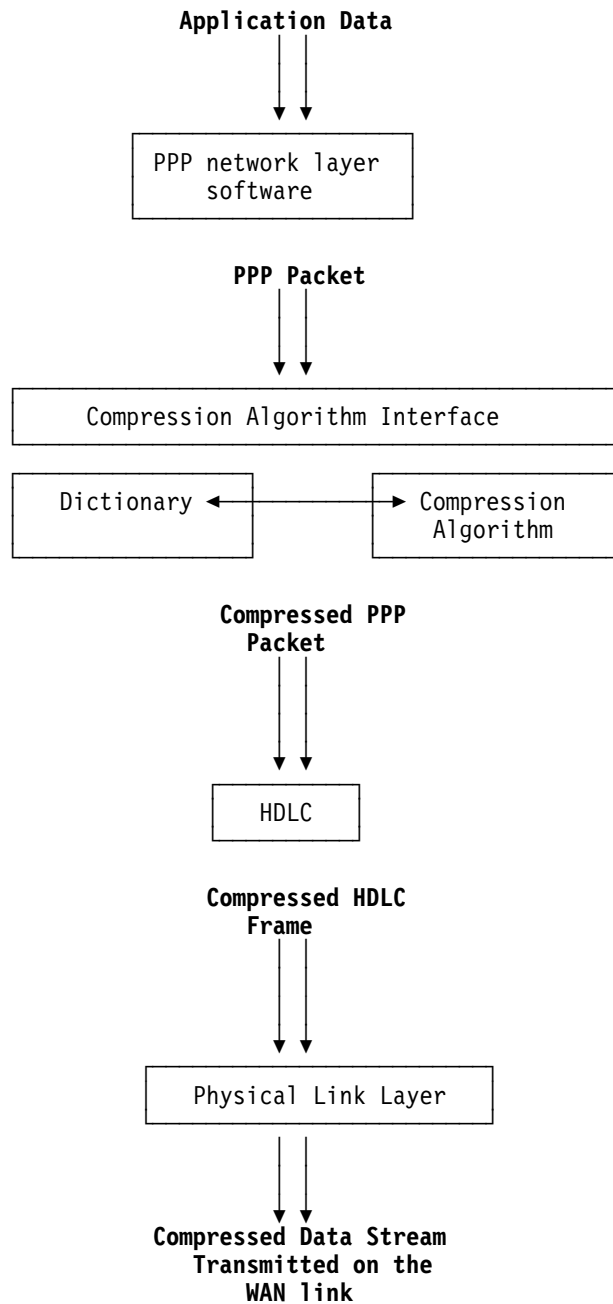
Figure 30-1. Data Compression on the Router

Compressing Data

The router performs PPP network layer encapsulation and then hands the data stream to the compression algorithm interface. The interface manipulates the PPP packet for compression by the compression algorithm.

The compression algorithm uses its compression dictionary to carry out the compression of the data stream to be transmitted on the WAN link (or decompression of data received on the WAN).

The compression process is reversed at the peer router on the WAN link. The data-link frame moves to the peer router. There the negotiated compression

algorithm and the compressor algorithm interface decompresses the data in the peer network layer and passes it to the PPP software.

Interaction with Bandwidth Reservation

Data compression interoperates with the Bandwidth Reservation System (BRS) for PPP interfaces. Data compression takes place only after BRS reorders or drops packets. See Chapter 10, "Using and Configuring Bandwidth Reservation (BRS) and Priority Queuing" on page 10-1 for details about bandwidth reservation.

Configuring and Monitoring Data Compression

Table 30-1 lists the data compression commands available at the PPP configuration prompt (`PPP Config>`).

Table 30-1. Data Compression Configuration Commands

Data Compression Command	Action
disable ccp	Disables data compression.
enable ccp	Enables data compression.
list ccp	Displays CCP options.
set ccp	Sets CCP options.

Notes:

1. The default for data compression is no data compression.
2. You cannot use Quick Config to enable or disable data compression.

At the PPP monitoring prompt (`PPP>`) use the **list ccp**, **list control ccp**, and **list cdp** commands to display the current state of the CCP software and list statistics related to CCP and the compression algorithm.

Table 30-2. Data Compression Monitoring Commands

Data Compression Command	Display
list control ccp	Lists CCP state and negotiated options.
list ccp	Lists CCP packet statistics.
list cdp	Lists compressed datagram statistics.

For more details on these commands, see Chapter 28, "Using and Configuring Point-to-Point Protocol Interfaces" on page 28-1 and Chapter 29, "Monitoring Point-to-Point Protocol Interfaces" on page 29-1.

Chapter 31. Configuring SDLC Relay

This chapter describes the Synchronous Data Link Control (SDLC) Relay configuration commands. The chapter includes the following sections:

- "Accessing the SDLC Relay Configuration Environment"
- "Basic Configuration Procedure"
- "SDLC Relay Configuration Commands" on page 31-2

For further information on when to use DLSw SDLC versus SDLC Relay, refer to "Relationship to the SDLC Relay Function" in the "Using and Configuring DLSw" chapter of *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 1 Release 1*.

Accessing the SDLC Relay Configuration Environment

To access the SDLC relay (SRLY) configuration environment:

1. At the `Config>` prompt, enter **set data-link srly**.

2. Enter the interface number.

3. To configure the SRLY interface, enter the **network** *interface#* command. The `SRLY` *interface#* `Config>` prompt is displayed when **network *interface#*** is entered:

   ```
   Config>network 2
   SDLC relay interface user configuration
   SRLY 1 Config>
   ```

4. To configure the SRLY protocol parameters, enter the **protocol sdlc** command. The `SDLC Relay config>` prompt is displayed when **protocol sdlc** is entered:

   ```
   Config>protocol sdlc
   SDLC Relay protocol user configuration
   SDLC Relay config>
   ```

Basic Configuration Procedure

This section outlines the minimum configuration steps required to get the SDLC Relay protocol up and running. For further configuration information and explanation, refer to the configuration commands described in this chapter.

Note: You must restart the router for new configuration changes to take effect.

- *Adding a number.* You must add a number to a group of primary or secondary ports using the **add group** command. The default number for this command is 1.

- *Adding a local port.* This identifies the interface that you are using for the local port. This also assures that no IP address is configured for the interface that you select. Use the **add local-port** command.

- *Adding a remote port.* This identifies the port directly connected to the remote side of the serial line. Use the **add remote-port** command.

31-1

SDLC Relay Configuration Commands

This section summarizes and then explains the SDLC Relay configuration commands. Both the **network** and **protocol** parameters for SDLC relay are documented in this chapter.

The SDLC Relay configuration commands allow you to specify router parameters for interfaces transmitting SDLC Relay frames. Restart the router to activate the configuration commands. Table 31-1 shows the commands for both the **network sdlc** and **protocol sdlc**.

Table 31-1. SDLC Relay Configuration Commands Summary

Command	Network SRLY	Protocol SDLC	Function
? (Help)	yes	yes	Lists all of the SDLC Relay configuration commands or lists the options associated with specific commands.
Add		yes	Adds groups, local ports, and remote ports.
Delete		yes	Deletes groups, local ports, and remote ports.
Disable		yes	Disables groups and ports.
Enable		yes	Enables groups and ports.
List	yes	yes	Displays entire SDLC Relay and group specific configurations.
Set	yes		Sets the link parameters and remote station parameters.
Exit	yes	yes	Exits the SDLC Relay configuration environment and returns to the CONFIG environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
Add
Delete
Disable
Enable
List
Set
Exit
```

Add

Use the **add** command to add group numbers, local ports, and remote ports.

Syntax: add   g̲roup
               l̲ocal-port
               r̲emote-port

group
   Assigns a number to a group of primary or secondary ports added to the
   router.

   Example: `add group`

   ```
   Group number: [1]? 1
   ```

   Group number   The group number that you are designating for the port.

local–port
   Identifies the interface that you are using for the local port.

   Example: `add local–port`

   ```
   Group number: [1]? 1
   Interface number: [0]? 2
   (P)rimary or (S)econdary:[S]? p
   ```

   Group number   The group number for the port. This number must
                           match one of the **add group** parameters configured
                           previously.

   Interface number   The interface number of the router that designates the
                           local port.

   Primary or Secondary Designates the port type, primary (P) or secondary
                           (S).

remote–port
   Identifies the IP address of the port directly connected to the serial line on the
   remote router.

   Example: `add remote–port`

   ```
   Group number: [1]? 1
   IP address of remote router:[0.0.0.0]? 128.185.121.97
   (P)rimary or (S)econdary:[S]? s
   ```

   Group number   The group number for the port. This number must
                           match one of the add group parameters configured
                           previously.

   IP address of remote router
                           Identifies the IP address of the interface on the remote
                           router.

   Primary or Secondary Designates the port type, primary (P) or secondary
                           (S).

Delete

Use the **delete** command to remove group numbers, local ports, and remote ports.

Syntax: <u>d</u>elete   group . . .
                       <u>l</u>ocal-port . . .
                       <u>r</u>emote–port

group *group#*
   Removes a group (group#) of SDLC Relay configured ports.

   Example: `delete group 1`

local–port *interface#*
   Removes the local port for the specified interface (interface#).

   Example: `delete local–port 2`

remote–port
> Removes the remote port for the specified group.

> **Example: `delete remote–port`**

>> ```
>> Group number: [1]? 1
>> (P)rimary or (S)econdary:[S]? S
>> ```

> *Group number*   The group number for the remote port.

> *Primary or Secondary* Designates the port type, primary (P) or secondary (S).

Disable

Use the **disable** command to suppress relaying for an entire relay group or a specific relay port.

Syntax: disable group . . .
>>>>> port

group *group#*
> Suppresses transfer of SDLC Relay frames to or from a specific group (group#).

> **Example: `disable group 1`**

port
> Suppresses transfer of SDLC Relay frames to or from a specific local port.

> **Example: `disable port`**

>> ```
>> Group number: [1]? 2
>> (P)rimary or (S)econdary:[S]? s
>> ```

> *Group number*   The group number of the port that you want to disable.

> *Primary or Secondary* Designates the port type, primary (P) or secondary (S).

Enable

Use the **enable** command to turn on data transfer for an entire group or a specific local interface port.

Syntax: enable group . . .
>>>>> port

group *group#*
> Allows transfer of SDLC Relay frames to or from the specified group (group#).

> **Example: `enable group 1`**

port
> Allows transfer of SDLC Relay frames to or from the specified local port.

> **Example: `enable port`**

>> ```
>> Group number: [1]? 2
>> (P)rimary or (S)econdary:[S]? s
>> ```

> *Group number*   The group number of the port that you want to enable.

> *Primary or Secondary* Designates the port type, primary (P) or secondary (S).

List (for network SRLY)

Use the **list** command to display the configuration of a specific group or of all groups.

Syntax: list

Example: `list`

```
Maximum frame size in bytes = 2048
Encoding:  NRZ
Idle State:  Flag
Clocking:  External
Cable Type:  RS-232 DTE
Speed (bps):  0
Transmit Delay Counter:  0
```

Maximum frame size in bytes

Maximum frame size that can be sent over the link.

Encoding   The transmission encoding scheme for the serial interface. Scheme is NRZ (non-return to zero) or NRZI (non-return to zero inverted).

Idle State   The data link idle state: flag or mark.

Clocking   The type of clocking: internal, external.

Cable Type   The serial interface cable type.

Speed (bps)   Lists the speed of the transmit and receive clocks.

Transmit Delay Counter

Number of flags sent between consecutive frames.

List (for protocol SDLC)

Use the **list** command to display the configuration of a specific group or of all groups.

Syntax: list   all

               group . . .

all

    Displays the configurations of all local ports.

Example: `list all`

```
            SDLC Relay Configuration

Group Number     Port Status      Net  SDLC Station   IP Address
                                  Number address (hex)
-------------- ---------------- ------ ------------- ------------
    1 (E)      Local   PRMRY (D)  2
    1 (E)      Remote  SCNDRY (E)                     128.185.452.11
    2 (D)      Local   PRMRY (D)  1
    2 (D)      Remote  SCNDRY (D)                     128.185.450.31
```

Group Number   Indicates the group number and the status of the group, enabled (E) or disabled (D).

Port Status   Indicates the type of port (local/remote primary/secondary) and its status, enabled (E) or disabled (D).

Net Number   Indicates the device number of the local port. This number matches the number displayed using the Config list devices command.

IP Address   Indicates the IP address of the remote port.

group *group#*
> Displays the configuration of a specified group.

Example: `list group 1`

```
            SDLC Relay Configuration

Group Number     Port Status      Net  SDLC Station   IP Address
                                  Number address (hex)
-------------- ---------------- ------ ------------- ------------
    1 (E)      Local   PRMRY (D)  2
    1 (E)      Remote  SCNDRY (E)                     128.185.452.11
```

Group Number   Indicates the group number and the status of the group, enabled (E) or disabled (D).

Port Status   Indicates the type of port (local/remote primary/secondary) and its status, enabled (E) or disabled (D).

Net Number   Indicates the device number of the local port. This number matches the number displayed using the Config list devices command.

IP Address   Indicates the IP address of the remote port.

Set

Use the **set** command to configure the SRLY parameters.

Syntax: set   cable
                       clocking
                       encoding
                       frame-size
                       idle
                       speed
                       transmit-delay

cable
> Sets the cable used on the serial interface. The options are:

- RS-232 DTE
- RS-232 DCE
- V35 DCE
- V35 DTE
- V36 DCE
- V36 DTE
- X21 DCE
- X21 DTE

Table 31-2 lists the cable types you can configure on the various adapters.

Table 31-2. Cable Types for 2216 Interfaces	
Adapter Type	**Cable Type**
8-port EIA 232	RS-232 DTE and RS-232 DCE
6-port V.35/V36	V.35 DCE, V.35 DTE, V.36 DCE, or V.36 DTE
8-port X.21	X.21 DCE and X.21 DTE

Example: `set cable V35 dte`

A DTE cable is used when you are attaching the router to some type of DCE device (for example, a modem or a DSU/CSU).

A DCE cable is used when the router is acting as the DCE and providing the clocking for direct attachment.

clocking *internal* or *external*
Configures the SRLY link's clocking. To connect to a modem or DSU, configure clocking as external. To connect directly to another DTE device, use a DCE cable, set the clocking to internal, and configure the clock speed. Use Table 31-4 on page 31-8 to determine the clock speeds you can set for the various adapters when internal clocking is used.

Example: `set clocking internal`

encoding *nrz* or *nrzi*
Configures the SRLY interface's encoding scheme as NRZ (Non-Return to Zero) or NRZI (Non-Return to Zero Inverted). NRZ is the default.

Example: `set encoding nrz`

frame-size
Configures the maximum size of the frames that can be transmitted and received on the data link. If this value is set to a larger value than that specified with the add remote-secondary command, then this value is changed to reflect that maximum. The 2216 generates an ELS message warning the user that this value is changing. The user will continue receiving this ELS message until it is changed in the SRAM configuration. Valid entries are shown in Table 31-3.

Table 31-3. Valid Values for Frame Size in Set Frame-Size Command

Minimum	Maximum	Default
576	8187	2048

Example: `set frame-size 4096`

idle flag
> Configures the transmit idle state for framing on the SRLY interface. The
> default is the flag option which provides continuous flags (7E hex) between
> frames.

> **Example:**   `set idle flag`

> The link will receive a flag idle transparently.

 idle mark
> Configures the transmit idle state for framing on the SRLY interface. The mark
> option puts the line in a marking state (OFF, 1) between frames.

> **Example:**   `set idle mark`

> The link will receive a mark idle transparently.

speed
> For internal clocking, this command specifies the speed of the transmit and
> receive clock lines. Use Table 31-4 to determine the link speeds you can set
> for the various adapters.

Table 31-4. Line Speeds When Internal Clocking is Used for 2216 Interfaces	
Adapter Type	**Speed Range**
8-port EIA 232	9600 to 64 000 bps
6-port V.35/V.36	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps
8-port X.21	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps

> For external clocking, this command does not affect the hardware. See
> Table 31-5 for the line speeds supported when external clocking is used.

Table 31-5. Line Speeds When External Clocking is Used for 2216 Interfaces	
Adapter Type	**Speed Range**
8-port EIA 232	2400 to 64 000 bps
6-port V.35/V.36	2400 to 2 048 000 bps
8-port X.21	2400 to 2 048 000 bps

> **Example:**   `set speed 64000`

transmit-delay *value*
> Allows the insertion of a delay between transmitted packets. This command
> ensures a minimum delay between frames so that it is compatible with older,
> slower serial devices at the other end. This value is specified as the number
> of flag bytes that should be sent between consecutive frames. The range is 0
> - 15. The default is 0.

> **Example:**   `set transmit-delay 6`

Exit

Use the **exit** command to exit the SDLC Relay configuration process and return to the CONFIG environment.

Syntax: <u>e</u>xit

Example: `exit`

Chapter 32. Monitoring SDLC Relay

This chapter describes how to use the Synchronous Data Link Control (SDLC) Relay console commands. The chapter includes the following sections:

- "Accessing the SDLC Relay Console Environment"
- "SDLC Relay Console Commands"
- "SDLC Relay Interfaces and the GWCON Interface Command" on page 32-4

Accessing the SDLC Relay Console Environment

To monitor information related to the SDLC Relay interface, access the interface console process by doing the following:

1. Enter the **status** command to find the PID for GWCON. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter the **talk** command and the PID for GWCON. For example:

   ```
   * talk 5
   +
   ```

   The GWCON prompt (+) is displayed on the console. If the prompt does not appear when you first enter GWCON, press **Return** again.

3. At the GWCON prompt, enter the **configuration** command to see the protocols and networks for which the router is configured. For example:

   ```
   + configuration
   ```

   See page 6-4 for more sample output from the **configuration** command.

4. Enter the **protocol sdlc** command. For example:

   ```
   + prot sdlc
   SDLC Relay>
   ```

   The SDLC Relay prompt is displayed on the console. You can then view information about the SDLC Relay ports by entering the SDLC Relay console commands.

SDLC Relay Console Commands

This section summarizes and then explains the SDLC Relay console commands. The SDLC Relay console commands allow you to view parameters for interfaces transmitting SDLC Relay frames. The SDLC Relay> prompt is displayed for all SDLC Relay console commands. Table 32-1 on page 32-2 shows the commands.

Table 32-1. SDLC Relay Console Commands Summary	
Command	**Function**
? (Help)	Lists all the SDLC Relay console commands or lists the options associated with specific commands.
Clear-Port-Statistics	Clears SDLC Relay statistics for the specified port.
Disable	Temporarily suppresses groups and ports.
Enable	Temporarily turns on groups and ports.
List	Displays entire SDLC Relay and group specific configurations.
Exit	Exits the SDLC Relay console process and returns to the GWCON environment.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
Clear-port-statistics
Disable
Enable
List
Exit
```

Example: **list ?**

```
all
group
```

Clear-Port-Statistics

Use the **clear-port-statistics** command to discard the SDLC Relay statistics for all ports. The statistics include counters for packets forwarded and packets discarded.

Syntax: c̲lear-port-statistics

clear-port-statistics
>   Clears port statistics gathered since the last time you restarted the router or cleared statistics.

>   **Example:**   **clear-port-statistics**

>    ```
Clear all port statistics? (Yes or No): Y
```

## Disable

Use the **disable** command to suppress data transfer for an entire group or a specific relay port.  SRAM (static read access memory) does not permanently store the effects of the **disable** console command.  Therefore when you restart the router, the effects of this command are erased.

**Syntax:**  d̲isable      g̲roup . . .
                         p̲ort

group  *group#*
>    Suppresses transfer of SDLC Relay frames to or from a specific group (group#).

**Example:** `disable group 1`

port *interface# primary-or-secondary*
   Suppresses transfer of SDLC Relay frames to or from a specific local port.

**Example:** `disable port`

```
Interface number: [0]? 2
(P)rimary or (S)econdary: [s]? P
```

Interface number   Indicates the interface number of the local port that you want to disable.

Primary or Secondary   Indicates whether the port is a primary or secondary.

Enable

Use the **enable** command to turn on data transfer for an entire group or a specific local interface port. SRAM does not permanently store the effects of the **enable** console command. Therefore when you restart the router, the effects of this command are erased.

Syntax: <u>en</u>able   group . . .
                        <u>p</u>ort

group *group#*
   Allows transfer of SDLC Relay frames to or from the specified group (group#).

Example: `enable group 1`

port
   Allows transfer of SDLC Relay frames to or from the specified local port.

Example: `enable port`

```
Interface number: [0]? 2
(P)rimary or (S)econdary: [s]? P
```

Interface number   Indicates the interface number of the local port that you want to enable.

Primary or Secondary   Indicates whether the port is a primary or secondary.

List

Use the **list** command to display the configuration of a specific group or of all groups.

Syntax: <u>l</u>ist   <u>a</u>ll
                     <u>g</u>roup . . .

all
   Displays the configurations of all local ports.

Example: `list all`

```
                  SDLC Relay Configuration

      Group  Port Status      Net    Packets    IP Address
      Num                     Num   fwrd  disc

      1 (E)  Local PRMRY (E)   2    2880   57
      1 (E)  Remote SCNDRY (E)      4860   13    128.185.452.11
      2 (D)  Local PRMRY (D)   1     0      0
      2 (D)  Remote PRMRY (D)        0      0    128.185.450.31
```

Group Number   Indicates the group number and the status of the group, enabled (E) or disabled (D).

<table>
<tr><td>Port Status</td><td>Indicates the type of port (local/remote primary/secondary) and its status, enabled (E) or disabled (D).</td></tr>
<tr><td>Net Number</td><td>Indicates the device number of the local port. This number matches the number displayed using the <code>Config></code> list devices command.</td></tr>
<tr><td>Packets (fwrd and disc)</td><td>Indicates how many packets were forwarded (fwrd) and discarded (disc) for that port.</td></tr>
<tr><td>IP Address</td><td>Indicates the IP address of the remote port.</td></tr>
</table>

group *group#*

Displays the configurations of a specified group.

Example: `list group 1`

```
                    SDLC Relay Configuration

      Group  Port Status      Net  Packets    IP Address
      Num                      Num  fwrd disc

      1 (E)  Local PRMRY (D)   2    2880   57
      1 (E)  Remote SCNDRY (E)      4860   13  128.185.452.11
```

Exit

Use the **exit** command to exit the SDLC Relay console process and return to the GWCON environment.

Syntax: <u>ex</u>it

Example: `exit`

SDLC Relay Interfaces and the GWCON Interface Command

While SDLC Relay interfaces have their own console processes for monitoring purposes, the router also displays complete statistics for installed network interfaces when you use the **interface** command from the GWCON environment. (For more information on the **interface** command, refer Chapter 6, "The GWCON (Monitoring) Process and Commands.")

Chapter 33. Configuring SDLC Interfaces

This chapter describes the SDLC configuration commands and includes the following sections:

- "Accessing the SDLC Configuration Environment"
- "Basic Configuration Procedure"
- "SDLC Configuration Requirements" on page 33-2
- "SDLC Configuration Commands" on page 33-2

You enter SDLC configuration commands at the `SDLC # Config>` prompt, where # identifies the interface you specify with the network command. Changes made to the routers configuration do not take effect immediately, but become part of the router's static configuration memory when it is restarted.

Accessing the SDLC Configuration Environment

Use the CONFIG process to change the configuration of the router. The new configuration takes effect when the router is restarted.

To enter the configuration process:

1. Enter **talk 6** (or **t 6**), at the OPCON (*) prompt. This brings you to the `CONFIG>` prompt as shown in the following example:

   ```
   MOS Operator Control

   * talk 6

   CONFIG>
   ```

   If the `CONFIG>` prompt does not appear immediately, press the **Enter** key again. All SDLC configuration commands are entered at the `SDLC config>` prompt.

2. At the `Config>` prompt, enter the **set data-link sdlc** command. When prompted, enter the name of the interface to associate with the SDLC device.

   ```
   Config>set data-link sdlc
   Interface number [0]? 2
   Config>
   ```

3. Next, enter the **network** command, plus the number of an SDLC interface that you entered earlier.

   ```
   Config>network 2
   SDLC 2 Config>
   ```

Refer to Chapter 1, "Getting Started (Introduction to the User Interface)" on page 1-1 for information related to the configuration environment.

Basic Configuration Procedure

This section outlines the minimum configuration required for SDLC to be usable by DLSw or by APPN.

Before beginning any configuration procedure, use the **list device** command from the config process to list the interface numbers of different devices. At the config prompt, select the interface you want to configure by entering either: **network** *interface number* or **n** *interface number*. If you need any further configuration

command explanations, refer to the configuration commands described in this chapter.

SDLC Configuration Requirements

In addition to the SDLC-specific configuration procedures and commands described in this chapter, you need to configure SDLC in the DLSw or APPN protocol. Only one protocol at a time, DLSw or APPN, may run over a given SDLC interface. In other words, link stations on a given SDLC interface cannot be divided between APPN and DLSw. If a DLSw and an APPN configuration exists for the same SDLC interface, the first protocol to come active will own the SDLC interface.

SDLC Configuration Commands

The SDLC configuration commands allow you to create or modify the SDLC interface configuration. This section summarizes and describes the commands you can issue from the `SDLC Config>` prompt within the network configuration console. Defaults for any command and its parameters are displayed on the console, they are enclosed in brackets immediately following the prompt.

Note: In addition to configuring SDLC using the commands described in this chapter, you also need to configure SDLC in the DLSw or APPN protocol.

2216 supports SDLC connections over RS-232, X.21, and V.35 serial interfaces.

Table 33-1 lists SDLC configuration commands and their function.

Table 33-1. SDLC Configuration Commands Summary	
Command	**Function**
? (Help)	Lists the configuration commands or lists any parameters associated with that command.
Add	Adds an SDLC end station.
Delete	Removes an SDLC end station.
Disable	Prevents connections to one of the SDLC link stations.
Enable	Allows connections to one of the SDLC link stations.
List	Displays configured information for one of the SDLC link stations.
Set	Configures specific interface and link-station information.
Exit	Exits the `SDLC config>` process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
Set
Add
Disable
Delete
Enable
List
Exit
```

Add

Use the **add** command to add an end station. The router is, by default the primary
end station. If you do not use this command and if you configured an SDLC station
in DLSw or in APPN, the end station is added for you. The software assigns the
following defaults to the station:

- Maximum BTU is maximum allowable by the interface
- Tx and Rx Windows are 7 for MOD 8, 127 for MOD 128

If the defaults are satisfactory, you do not need to add SDLC station.

Syntax: <u>a</u>dd   station
Example: **add station**

```
Enter station address (in hex) [C3]?
Enter station name [SDLC_C3]?
Include station in group poll list ([Yes] or No):
Enter max packet size [2009]?
Enter receive window [7]?
Enter transmit window [7]?
```

Enter station address	The station's SDLC address in the range 01 - FE.
Enter station name	The name designation of the SDLC station (maximum characters is 8).
Include station in group poll list	Select whether or not to include this station in the group poll list for this link. The SDLC software supports the IBM 3174 group poll function for SDLC secondary station. You must add a group poll address using the **set link group-poll** command for this parameter to have an affect.
Enter max packet size	The maximum packet size that can be sent to or received from the remote link station. This value cannot be greater than that specified for the link. This value is configured with the **set link frame-size** command.
Enter receive window	The maximum number of packets that the router can receive without sending a response.
Enter transmit window	The maximum number of packets that the router can transmit without receiving a response.

Delete

Use the **delete** command to remove the specified end station (station name or
address) from the SDLC configuration. The router is considered the primary end
station (default).

Syntax: <u>de</u>lete   station *name* or *address*

Example: **delete station c1**

Disable

Use the **disable** command to prevent connections from being created with a SDLC link station.

Syntax: <u>dis</u>able   <u>link</u>
               <u>s</u>tation . . .

link

Prevents the transmitting and receiving of data to all configured SDLC link stations on the interface.

Example: `disable link`

station *name* or *address*

Prevents the transmitting and receiving of data to the specified end station (station name or address).

Example: `disable station c1`

Enable

Use the **enable** command to enable connections to remote SDLC link stations.

Syntax: <u>en</u>able   <u>link</u>
               <u>s</u>tation

link

Allows subsystems in the router (for example, DLSw) to use SDLC's facilities.

Example: `enable link`

station *name* or *address*

Allows connections to the specified secondary remote end station (link station name).

Example: `enable station c1`

List

Use the **list** command to display configuration information on one or all SDLC link stations.

Syntax: <u>l</u>ist   <u>link</u>
                <u>s</u>tation *name* or all

link

Displays the SDLC interface's configuration.

Example: `list link`

```
Link configuration for: LINK_2   (ENABLED)

Role:       SECONDARY      Type:        POINT-TO-POINT
Duplex:     FULL           Modulo:      8
Idle state: FLAG           Encoding:    NRZ
Clocking:   EXTERNAL       Frame Size:  2048
Speed:      0              Group Poll:  F3
Cable       V.36 DTE

Timers:     XID/TEST response:  2.0 sec
            SNRM response:      2.0 sec
            Poll response:      0.5 sec
            Inter-poll delay:   0.2 sec
            Inter-frame delay:  DISABLED
            Inactivity timeout  30.0 sec
```

```
Counters:  XID/TEST retry:  8
           SNRM retry:      6
           Poll retry:      10
```

Link configuration	The name and status of SDLC link station that are in the router's configuration.
Role	The primary, secondary, or negotiable role for link stations that you configure using the **set link role** command.
Type	The type of link, MULTIPOINT or POINT-TO-POINT.
Duplex	Duplex configuration, HALF or FULL.
Modulo	The sequence number range to use on the link: MOD 8 (0-7) or MOD 128 (0 - 127).
Idle state	The bit pattern (FLAG or MARK) transmitted on the line when the interface is not transmitting data.
Speed	The physical data rate of the interface. When the clocking is internal, this is the data rate generated by the internal clock.
Group Poll	Address used for the group poll feature for multipoint link configurations. Secondary stations having group inclusion coded as yes will respond to unnumbered polls received from this address. This address must be non-null for the group poll feature to be in effect for any secondary stations under this link. Each secondary station will still have a unique station address in addition to the group address.
Cable	Specifies the type of cable in use (RS-232, V.35, V.36, or X.21).
Encoding	Configures the SDLC transmission encoding scheme as NRZ (Non-Return to Zero) or NRZI (Non-Return to Zero Inverted).
Clocking	Interface clocking, EXTERNAL or INTERNAL.
Frame Size	The maximum frame size that can be sent over the interface.
Timers:	All the timers listed below have a 100ms resolution.
XID/TEST resp.	The time to wait for an XID or TEST response message before retransmitting the XID or TEST frame. A value of 0 indicates that the router will continue to retry indefinitely.
SNRM response	The maximum time to wait for an UA response message before the station retransmits SNRM(E).
Poll response	The maximum time to wait for a response from any polled station before retrying.
Inter-poll delay	The amount of time the router (configured with a primary role) waits after receiving a response, before polling the next station.
Interframe delay	The number of flags sent between frames.
Inactivity timeout	For idle NRM/E secondary stations, sets the time after which the interface changes the station to its recovery state. A 0 (zero) causes the station to remain idle indefinitely.
Counters:	
XID/TEST retry	The maximum number of times the router sends an XID or TEST frame without receiving a response before timing out. A value of 0 indicates that the router will retry indefinitely.
SNRM	The maximum number of times the router will send an SNRM(E) frame without receiving a response before timing out. A value of 0 indicates that the router will retry indefinitely.

Poll retry		The maximum number of times the router polls the station without receiving a response before timing out. A value of 0 indicates that the router will continue to retry indefinitely.

station *all* or *address* or *link station name*

Displays information for the specified SDLC link station or for all link stations.

Example: `list station c1`

```
Address   Name    Status   Max BTU  Rx Window  Tx Window
-------   ------   ------   -------  ---------  ---------
C1(00)  SDLC_C1   Enabled    2005       7          7
```

Example: `list station all`

```
Address   Name      Status    Max BTU  Rx Window  Tx Window
-------  --------  ----------  -------  ---------  ---------
C1(00)  SDLC_C1    ENABLED      2005       7          7
C3(F3)  SDLC_C3    DISABLED     2009       7          7
```

Address	The address of the SDLC link station. The address in parentheses is the group address of the station. A (00) indicates that a group address is not defined.
Name	The character string name designation of SDLC link station.
Status	The status of the SDLC link station, ENABLED or DISABLED.
Max BTU	The frame size limit of the station. This frame size must not be larger than the maximum Basic Transmission Unit (BTU) packet size configured with the **set link frame-size** command.
Rx Window	The size of the receive window.
Tx Window	The size of the transmit window.

Set

Use the **set** command to configure specific information for one or all SDLC link stations.

Syntax: s̲et   l̲ink c̲able
                l̲ink c̲locking
                l̲ink d̲uplex . . .
                l̲ink e̲ncoding . . .
                l̲ink f̲rame-size
                l̲ink g̲roup poll ...
                l̲ink i̲dle . . .
                l̲ink i̲nactivity ...
                l̲ink inter-frame delay
                l̲ink m̲odulo . . .
                l̲ink n̲ame
                l̲ink p̲oll . . .
                l̲ink r̲ole . . .
                l̲ink s̲nrm
                l̲ink s̲peed
                l̲ink t̲ype . . .
                l̲ink x̲id/test
                s̲tation *address* . . .

link cable *type*

Sets the cable connected to this interface. The options are the following DCE and DTE types: V.36, RS-232, V.35, and X.21.

Table 33-2 lists the cable types you can configure on the various adapters.

Table 33-2. Cable types for 2216 Interfaces

Adapter Type	Cable lype
8-port EIA 232	RS-232 DTE and RS-232 DCE
6-port V.35/V36	V.35 DCE, V.35 DTE, V.36 DCE, or V.36 DTE
8-port X.21	X.21 DCE and X.21 DTE

A DTE cable is used when you are attaching the router to some type of DCE device (for example, a modem or a DSU/CSU).

A DCE cable is used when the router is acting as the DCE and providing the clocking for direct attachment.

Example: `set link cable V35 dte`

link clocking *internal* or *external*

Configures the SDLC link's clocking. To connect to a modem or DSU, set clocking external. To connect directly to another DTE device, use a DCE cable, set the clocking to internal, and configure the clock speed. Use Table 33-4 on page 33-9 to determine the clock speeds you can set for the various adapters when internal clocking is used.

Example: `set link clocking internal`

link duplex *full* or *half*

Configures the SDLC line for full-duplex or half-duplex.

Example: `set link duplex full`

link encoding *nrz* or *nrzi*

Configures the SDLC transmission encoding scheme as NRZ (Non-Return to Zero) or NRZI (Non-Return to Zero Inverted). NRZ is the default.

Example: `set link encoding nrz`

link frame-size

Configures the maximum size of the frames that can be transmitted and received on the data link. Valid entries are shown in Table 33-3.

Table 33-3. Valid Values for Frame Size in Link Frame-Size Command

Minimum	Maximum	Default
576	8187	2048

Set the link frame size greater than the maximum packet size that you configured with the **set station xxx max packet** command. Otherwise, the router automatically resets the maximum packet size to the link frame size and issues the following ELS message:

`SDLC.054: nt 3 SDLC/0 Stn xx-MaxBTU too large for Link adjusted (4096->2048)`

Example: set link frame-size

`Frame size in bytes (576 - 18000)[2048]?`

Example: set link frame-size

```
Frame size in bytes (576 - 8187)[2048]?
```

link group-poll
Sets a group poll address for secondary stations on the link. The SDLC
software supports the IBM 3174 group poll function. Use the **add station** or
the **set station group-inclusion** command to include a station in the group
poll list.

Example: set link group-poll

```
        Enter group poll address (in hex) [00:]?f3
        Group poll support enabled
```

link idle flag
Configures the transmit idle state for SDLC framing. The default is the flag
option which provides continuous flags (7E) between frames.

Example: set link idle flag

The link will receive a flag idle transparently.

link idle mark
Configures the transmit idle state for SDLC framing. The mark option puts the
line in a marking state (OFF, 1) between frames.

Example: set link idle mark

link inactivity *#-of-seconds*
For idle NRM/E secondary stations, sets the time after which the interface
changes the station to its recovery state. The range is 0 to 7200 seconds.
The default is 30. A 0 (zero) causes the station to remain idle indefinitely.

Example: set link inactivity

```
Enter secondary link station inactivity timeout :[30.0]?
```

link inter-frame delay
Allows the insertion of a delay between transmitted packets. This command
ensures a minimum delay between frames so that it is compatible with older,
slower serial devices at the other end. The delay is specified in terms of the
number of flags that should be sent between consecutive frames. The range is
0 to 15 flags and 0 (in other words, no flags) is the default value.

Example: set link inter-frame delay

```
Transmit Delay Counter [0]?
```

link modulo 8 or 128
Specifies the sequence number range to use on the link: MOD 8 (0-7) or MOD
128 (0 - 127). Default is 8.

Note: When you change this value, the window sizes become invalid. Use
the **set station** command to change the receive window and transmit window
sizes. Valid window sizes for mod 8 are 0 through 7; for mod 128 they are 8
through 127.

Also, at connection start-up, an SNRME rather than a SNRM is used and
supervisory frame headers are expanded by an additional byte.

Example: set link modulo 8

link name
Establishes a character string for the link that you are configuring. This
parameter is for informational purposes only.

Example: `set link name`

`Enter link name: [LINK_0]?`

link poll delay

Configures the time delay between each poll that is sent over the interface.

Example: `set link poll delay`

`Enter delay between polls [0.2]?`

link poll retry

Configures the number of times the interface retries to poll the secondary SDLC link station before it closes the connection.

Example: `set link poll retry`

`Enter poll retry count (0 = forever) [10]?`

link poll timeout

Configures the amount of time the interface waits for a poll response before timing out.

Example: `set link poll timeout`

`Enter poll timeout [0.5]?`

link role *primary* or *secondary* or *negotiable*

Configures the interface as an SDLC primary, secondary, or negotiable link station (default is primary).

Example: `set link role primary`

link snrm *timeout* or *retry*

Configures the following SNRM(E) information for primary stations:

timeout	The time to wait for an Unnumbered Acknowledgements (UA) response before retransmitting an SNRM(E).
retry	The number of times to retransmit an SNRM(E) without receiving a response before giving up.

Example: `set link snrm timeout`

`Enter SNRM response timeout [2.0]?`

Example: `set link snrm retry`

`Enter SNRM retry count (0=forever) [6]?`

link speed

For internal clocking, this command specifies the speed of the transmit and receive clock lines. Use Table 33-5 on page 33-10 to determine the clock speeds you can set for the various adapters.

Table 33-4. Line Speeds When Internal Clocking is Used for 2216 Interfaces

Adapter Type	Speed range
8-port EIA 232	9600 to 64 000 bps
6-port V.35/V.36	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps
8-port X.21	9600 to 460 800 bps, 1 544 000 bps, or 2 048 000 bps

You can also set the line speed when clocking is external though this has no affect on the hardware. See Table 33-5 on page 33-10 for line speeds supported when external clocking is used.

Table 33-5. Line Speeds When External Clocking is Used for 2216 Interfaces

Adapter Type	Speed range
8-port EIA 232	2400 to 64 000 bps
6-port V.35/V.36	2400 to 2 048 000 bps
8-port X.21	2400 to 2 048 000 bps

Example: `set link speed`

```
Line Speed [64000]?
```

link type *multipoint* or *point-to-point*
Configures the SDLC link to either a multipoint link or a point-to-point link.

Example: `set link type multipoint`

link xid/test *timeout* or *retry*
Configures the following XID/test information for primary stations:

timeout	The maximum amount of time to wait for an XID or TEST frame response before retransmitting the XID or TEST frame.
retry	The maximum number of times an XID or TEST frame is resent before giving up. A 0 (zero) causes the router to retry indefinitely.

Example: `set link xid/test timeout 10`

remote-secondary *address* or *link_station_name address <argument>*
Changes the remote station's SDLC address in the range 02 - FE.

Example: `set remote-secondary SDLC_C1 address ce`

station *address* or *name* address
Changes the station's SDLC address in the range 01 to FE.

Example: `set station c1 address`

```
Enter station address (in hex) [C1]?
```

station *address* or *link station name* group-inclusion *no* or *yes*
For SDLC secondary stations, set whether to include this station in the group poll list for this link. For this to be effective, add a group poll address using the **set link group-poll** command.

Example: `set station c1 group-inclusion yes`

station *address* or *name* max-packet
The maximum size of the packet that the station can receive (default: 2048). Do not set the maximum packet size larger than the link frame size that is configured with the **set link frame-size** command; if you do, the router automatically resets the maximum packet size to the link frame size and issues the following ELS message:

```
SDLC.054: nt 3 SDLC/0 Stn xx-MaxBTU too large for Link adjusted (4096->2048)
```

Example: `set station c1 max-packet`

```
Enter max packet size [2048]?
```

station *address* or *name* name
> The name of the SDLC station.

> **Example: set station c1 name**

> Enter station name [SDLC_C1]?

station *address* or *name* receive window
> The maximum number of frames the router can receive before sending a response. The range is 1 to 7. The default is 7.

> **Example: set station c1 receive-window**

> Enter receive window [7]?

station *address* or *name* transmit-window
> The maximum number of frames the router can transmit before receiving a response frame. The range is 1 to 7. The default is 7.

> **Example: set station c1 transmit-window**

> Enter transmit window [7]?

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: exit

Chapter 34. Monitoring SDLC Interfaces

This chapter describes the SDLC console commands. Some of these commands are identical to those in the SDLC Config> process, and they allow you to dynamically configure the SDLC interface without permanently affecting the SRAM configuration.

This chapter includes the following sections:

Changes made at the configuration command console (SDLC CONFIG>) become part of the SRAM configuration when you restart the router.

Conversely, SDLC monitoring commands entered within the SDLC monitoring process take effect immediately. However, changes made with monitoring commands do not become part of the router's static configuration. When the router is restarted, the effects of the monitoring commands are overwritten by the router's static configuration. Monitoring consists of these actions:

- Monitoring the protocols and network interfaces that are currently in use by the router

- Making real-time changes to the SDLC configuration without permanently affecting the SRAM configuration

- Displaying ELS (Event Logging System) messages relating to router activities and performance

Accessing the SDLC Monitoring Environment

The monitoring environment is the GWCON process. To enter the GWCON process:

1. Enter **talk 5** (or **t 5**) at the OPCON (*) prompt. This brings you to the GWCON (+) prompt as shown in the following example:

   ```
   MOS Operator Control

   * talk 5
   +
   ```

2. Next, enter the **network #** command using the number that identifies the interface that you previously configured for the SDLC device.

   ```
   + network 2
   SDLC Console
   SDLC-2>
   ```

You enter all GWCON (Monitoring) commands at the + prompt.

Refer to Chapter 1, "Getting Started (Introduction to the User Interface)" on page 1-1 for information related to the monitoring environment.

SDLC Console Commands

This section summarizes and then explains the SDLC console and related commands. Use these commands to gather information from the database. Table 34-1 lists SDLC console commands and their function.

Command	Function
Table 34-1. SDLC Console Commands Summary	
Command	**Function**
? (Help)	Lists all the SDLC monitoring commands or any options associated with those commands.
Add	Adds an SDLC link station
Clear	Clears the counters on the SDLC interface.
Delete	Dynamically removes an SDLC link station.
Disable	Disables connections to one SDLC link station.
Enable	Enables connections to one SDLC link station.
List	Displays SDLC link stations configurations and link station information.
Set	Configures specific interface and link station information.
Test	Tests the link between the router and the SDLC link station.
Exit	Exits the SDLC console process.

? (Help)

Use the **?** (help) command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: **?**

```
ADD
CLEAR Counters
DELETE
DISABLE
ENABLE
LIST
SET
TEST
EXIT
```

Add

Use the **add** command to add an end station. The router is, by default the primary end station. If you do not use this command and if you configured an SDLC station in DLSw or APPN, the end station is added for you.

Syntax: <u>a</u>dd   <u>s</u>tation

For an example and for additional information on the **add** command, see "Add" on page 33-3.

Clear

Use the **clear** command to clear counters for the interface, for a station, or for all stations. Use the **list all stations** command to list stations.

Syntax: <u>c</u>lear   <u>li</u>nk

                <u>st</u>ation ...

link *name* or *address*

    Clears the counters for an SDLC interface.

    Example: `clear link`

link *name* or *address*

    Clears the counters for an SDLC interface.

    Example: `clear link c1`

station *name* or *address* or all

    Clears counters for a specific station or for all stations.

    Example: `clear station c1`

Delete

Use the **delete** command to terminate an existing SDLC connection without affecting the SDLC configuration in SRAM. This command terminates any SDLC session that may be in progress on the link station. The router is considered the primary end station by default.

Syntax: <u>de</u>lete   <u>s</u>tation *name* or *address*

Example: **`delete station c1`**

Disable

Use the **disable** command to disable connection establishment on one or all SDLC link stations without affecting the SDLC configuration in SRAM. The **disable** command also terminates any existing connection to the station.

Syntax: <u>dis</u>able   <u>li</u>nk

                <u>st</u>ation . . .

link

    Prevents connection on all configured SDLC link stations on the interface by terminating all connections.

    Example: `disable link`

station *name* or *address*

    Prevents connection to the specified end station (link station name) by terminating any existing connection.

    Example: `disable station C1`

Enable

Use the **enable** command to enable connection establishment with remote SDLC link stations without affecting the SDLC configuration SRAM.

Syntax: <u>en</u>able   <u>li</u>nk

                <u>st</u>ation . . .

link
    Allows subsystems (for example, DLSw) to use SDLC's facilities.

Example: `enable link`

<u>s</u>tation *name* or *address*
    Allows connections to the specified end station.

Example: `enable station c1`

List

Use the **list** command to display statistics specific to the data link layer and the interface.

Syntax: <u>l</u>ist   <u>li</u>nk <u>co</u>nfiguration
             <u>li</u>nk <u>co</u>unters
             <u>s</u>tation . . .

link configuration
    Displays information for all configured SDLC link stations on the interface.

Example: `list link configuration`

For an example and for additional information on the **list** command, see "List" on page 33-4.

link counters
    Displays information for the SDLC counters since the last router restart or the last clear counters.

Example: `list link counters`

```
       I-Frames    I-Bytes    Re-Xmit    UI-Frames    UI-Bytes
       --------    -------    -------    ---------    --------
Send      0           0          0           0           0
Recv      0           0                      0           0

          RR         RNR        REJ          UP
       -------     ------      ----      ---------
Send      0           0          0           0
Recv      0           0          0           0
```

I-Frames Total number of Information frames received and transmitted.

I-Bytes Total number of Information bytes received and transmitted.

Re-Xmit Total number of frames that were retransmitted.

UI-Frames Total number of Unnumbered Information frames received and transmitted.

UI-Bytes Total number of Unnumbered Information bytes received and transmitted.

RR Total number Receive-Ready (RRs) received and transmitted.

RNR Total number Receive-Not-Ready (RNRs) received and transmitted.

REJ Total number of Rejects received and transmitted.

UP Unnumbered Polls (group poll) received and transmitted.

station *all* or *address* or *link station name*
    Displays the status of the specified SDLC link station or all stations. The software displays an * next to the stations that were not explicitly configured

using the **add station** command but were added to the configuration because they were defined and activated in the protocol layer (DLSw or APPN).

Displays information for the specified SDLC link station (link station name) on the interface.

Example: `list station all`

```
Address    Name      Status    Max BTU   Rx Window   Tx Window
--------   -----     ------    -------   ---------   ---------
C1(00)     SDLC_C1   IDLE      2048      7           7
C2(F3)     SDLC_C2   ENABLED   2048      7           7
```

Example: `list station c1`

```
Address   Name     Status     Max BTU    Rx Window   Tx Window
-------   ------   --------    -------    ---------   ---------
* C1(00)  SDLC_C1  ENABLED     2048       7           7
```

Address
: The address of the SDLC link station. The address in parentheses is the group address of the station. A (00) indicates that a group address is not defined.

Name
: The character string name designation of SDLC link station.

Status
: The status of the SDLC link station:

Enabled	Enabled, but not allocated
Idle	Allocated, but not in use
Connected	Connected
Disconnected	Disconnected
Connecting	Connection establishment in progress.
Discnectng	Disconnection in progress
Recovering	Attempting to recover from a temporary data link error.

Max BTU
: The frame size limit of the remote station. This frame size must not be larger than the maximum Basic Transmission Unit (BTU) packet size configured with the **set link frame-size** command. The default is 2048 bytes.

Rx Window
: The size of the receive window.

Tx Window
: The size of the transmit window.

station *name* or *address* counters
: Displays frame transmit and receive counts for the specified link station.

Example: `list station c1 counters`

```
        I-Frames    I-Bytes    Re-Xmit    UI-Frames    UI-Bytes    XID-Frames
        ----------  ---------- ---------  ----------   ----------  ----------
Send           9        384          0           0           0           6
Recv          29      42792                      0           0           3

          RR         RNR        REJ         TEST         SNRM        DISC
        ----------  ---------- ---------  ----------   ----------  ----------
Send         598          0          0           0           1           0
Recv         587          0          0           0           0           0

          UA         DM         FRMR        UP
        ----------  ---------- ---------  ---------
Send           0          0          0           0
Recv           1          0          0           0
```

I-Frames
: Number of information frames received and transmitted

I-Bytes	Number of information bytes received and transmitted
Re-Xmit	Number of frames retransmitted
UI-Frames	Number of Unnumbered Information frames received and transmitted
UI-Bytes	Number of Unnumbered Information bytes received and transmitted
XID-Frames	Number of Exchange Identification frames received and transmitted
RR	Number of Receive Ready frames received and transmitted
RNR	Number of Receive Not Ready frames received and transmitted
REJ	Number of Rejects received and transmitted
TEST	Number of Test frames received and transmitted
SNRM	Number of Set Normal Response Mode frames received and transmitted
DISC	Number of Disconnect frames received and transmitted
UA	Number of Unnumbered Acknowledgement frames received and transmitted
DM	Number of Disconnected Mode frames received and transmitted
FRMR	Number of Frame Reject frames received and transmitted
UP	Unnumbered Polls (group poll) received and transmitted.

Set

Use the **set** command to dynamically configure specific information for one or all SDLC link stations without affecting the SRAM configuration. In the SDLC monitoring environment, the **set** command can be executed only on disabled links or stations. All time values are entered in seconds, with a 0.1 second resolution.

Syntax: set   link modulo . . .
                  link name
                  link poll . . .
                  link role . . .
                  link snrm(e)
                  link type . . .
                  link xid/test
                  station . . .

link modulo

Dynamically changes the range of sequence numbers to be used on the data link without affecting the SRAM configuration. Modulo 8 specifies a sequence number range 0 - 7, and modulo 128 specifies 0 - 127. Default is 8.

Note: When you change this value, the transmit and receive window sizes become invalid. Use the **set station** command to change the receive-window and transmit-window sizes.

Example: `set link modulo 8`

link name

Dynamically changes the name of the link without affecting the SRAM configuration. A maximum of 8 characters can be entered. This parameter is for informational purposes only.

Example: `set link name`

`Enter link name: [LINK_0]?`

link poll delay or timeout or retry

Dynamically changes the following poll information without affecting the SRAM configuration.

delay   Configures the delay between each poll that is sent over the interface.

timeout   Configures the amount of time the router waits for a poll response before timing out.

retry   Configures the number of times the interface retries to poll the remote SDLC link station before it closes the connection.

Example: `set link poll delay`

`Enter delay between polls [0.2]?`

link role *primary*, *secondary*, or *negotiable*

Configures the interface as an SDLC primary, secondary, or negotiable link station. The default is primary. Use of this command does not affect the SRAM configuration. Information removed 8-14-96. LAD

Example: `set link role primary`

link snrm *timeout* or *retry*

For primary stations, dynamically changes the following SNRM(E) information without affecting the SRAM configuration.

timeout   The time to wait for an Unnumbered Acknowledgement (UA) response before retransmitting an SNRM(E).

retry   The number of times to retransmit an SNRM(E) without receiving a response before giving up.

Example: `set link snrm timeout`

`Enter SNRM response timeout [2.0]?`

link type multipoint or point-to-point

Dynamically changes the SDLC link to either a multipoint link or a point-to-point link without affecting the SRAM configuration.

Example: `set link type multipoint`

link xid/test *timeout* or *retry*

For primary stations, dynamically changes the following XID/test information without affecting the SRAM configuration.

timeout   The maximum amount of time to wait for an XID or TEST frame response before retransmitting the test frame.

retry   The maximum number of times an XID or TEST frame is resent before giving up. A 0 (zero) causes the router to retry indefinitely.

Example: `set link xid timeout 10`

Note: Examples for, and explanations of, the following parameters can be found in the SDLC configuration chapter at "Set" on page 33-6.

station *address* or *name* address
>   Changes the station's SDLC address.

station *address* or *name* max-packet
>   Maximum size of packet that this station can receive.

station *address* or *name* name
>   Name of the SDLC station.

station *address* or *name* receive-window
>   Maximum number of frames router sends before responding.

station *address* or *name* transmit-window
>   Maximum number of frames router transmits before receiving a response frame.

Test

Transmits a specified number of TEST frames to the specified station and waits for a response. Use this command to test the integrity of the connection. Press any key to cancel the test.

Note: Disable the specified link station before using this command.

Syntax: <u>te</u>st <u>s</u>tation *name* or *address* *#frames-to-send* *frame-size*

Example: `test station c1`

```
Number of frames to send [1]? 5
Frame length [265]?
Starting echo test -- press any key to abort
5 frames sent, 5 frames received, 0 compare errors, 0 timeouts
```

Number of test frames to send
>   Total number of frames to send.

Frame length   Length of frames to be sent. Frame length cannot be larger than the maximum frame length of the specified station.

The test may be aborted by pressing any key.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: <u>e</u>xit

Example: `exit`

SDLC Interfaces and the GWCON Interface Command

While the SDLC interface has a console process for monitoring purposes, the 2216 also displays complete statistics for installed interfaces when you use the **interface** command from the GWCON environment. (For more information on the interface command, refer to Chapter 6, "The GWCON (Monitoring) Process and Commands.")

Statistics Displayed for SDLC Interfaces

Using the **interface** command, you can display statistics for SDLC devices without entering the SDLC monitoring process. To do this, enter the **interface** command and an interface number at the + prompt, as shown:

```
  + interface 12
```

This command lists statistics in the following format:

```
                                     Self-Test Self-Test Maintenance
Nt Nt'  Interface Slot-Port          Passed    Failed    Failed
12 12   SDLC/0     Slot: 8  Port: 2      2         1         0

  SDLC MAC/data-link on V.35/V.36 interface

  Adapter cable:             V.35 DTE

  V.24 circuit: 105 106 107 108 109
  Nicknames:    RTS CTS DSR DTR DCD
  PUB 41450:    CA  CB  CC  CD  CF
  State:        ON  ON  ON  ON  ON

  Line speed:            64.000 Kbps
  Last port reset:       1 hour, 20 minutes, 42 seconds ago

  Input frame errors:
   CRC error                      0    alignment (byte length)     0
   missed frame                 182    too long (>  2062 bytes)    0
   aborted frame                  0    DMA/FIFO overrun            0
  Output frame counters:
   DMA/FIFO underrun errors        0    Output aborts sent          0
```

Nt   Indicates the interface number as assigned by software during initial configuration.

Nt'   Indicates the interface number as assigned by software during initial configuration.

                    Note: For SDLC interfaces, the Nt' interface number is always the same as the Nt interface number.

Slot   Indicates the slot number of the interface that is running SDLC.

Port   Indicates the port number of the interface that is running SDLC.

Self-test passed   Indicates the total number of times the SDLC interface passed its self-test.

Self-test failed   Indicates the total number of times the SDLC interface was unable pass its self-test.

Maintenance failed Indicates the number of maintenance failures.

The following parameters are displayed only if a cable is connected. The information displayed depends on the cable that is connected. Different parameters are displayed with other cables.

Adapter cable	Indicates the type of adapter cable that the level converter is using.
V.24 circuit	Indicates the circuits being used on the V.24.
Nicknames	Indicates the signals being used on the V.24 circuit.
RS-232	The EIA 232 (RS 232) circuit names.
State	Indicates the state of V.24 circuits, signals, and pin assignments (ON or OFF).
Line speed (configured)	Indicates the currently configured line speed for the SDLC interface.
Last port reset	Indicates how long ago the port was last reset.
Input frame errors	Indicates the input frame error type (CRC error, too short, aborted, alignment, too long, DMA/FIFO overrun) and the total number of errors that have occurred.
Output frame counters	Indicates the total number of DMA/FIFO overruns and output aborts sent for output frames.
Missed frame	When a frame arrives at the device and there is no buffer available, the hardware drops the frame and increments the missed frame counter.
L & F bits not set	On serial interfaces, the hardware sets input-descriptor information for arriving frames. If the buffer can accept the complete frame upon arrival, the hardware sets both the Last and First bits of the frame, indicating that the buffer accepted the complete frame. If either of the bits is not set, the packet is dropped, the `L & F bits not set` counter is incremented, and the buffer is cleared for reuse.
	Note: It is unlikely that the `L & F bits not set` counter will be affected by traffic.

Chapter 35. Using and Configuring the V.25 bis Network Interface

The V.25 bis interface allows routers to establish serial connections over switched telephone lines using V.25 bis modems. This chapter describes how to configure a V.25 bis interface. It includes the following sections:

- "Accessing the Interface Configuration Process"
- "Before You Begin"
- "Configuration Procedures" on page 35-2
- "V.25 bis Configuration Commands" on page 35-5

Notes:

1. You can assign a destination name to a **connection list** and assign a destination number to each line in the list. When that destination name is called, the numbers in the list are tried one by one until a connection is made or the list is exhausted.

2. V.25bis is supported only on the 8-port EIA 232 adapter.

Accessing the Interface Configuration Process

Use the following procedure to access the V.25 bis configuration process.

1. At the OPCON prompt (*), enter the **status** command to find the PID for CONFIG. (See page 1-5 for sample output of the **status** command.)

2. At the OPCON prompt, enter the **talk** command and the PID for CONFIG. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

```
* talk 6
Config>
```

After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter **CONFIG**, press **Return** again.

3. At the CONFIG prompt, enter the **list devices** command to display the network interface numbers for which the router is currently configured.

4. Record the interface numbers.

5. Enter the CONFIG **network** command and the number of the interface you want to configure. For example:

```
Config> network 1
V.25bis Config>
```

The V.25 bis configuration prompt now displays on the console.

Before You Begin

Before you configure V.25 bis on the router, make sure you have the following:

- V.25 bis modems that support synchronous V.25 bis commands and the 1988 ITU/CCITT V.25 bis specification.

- If your modem does not automatically detect answer originate, you must:

             35-1

– Configure the modem at one end of the link to originate calls.
– Configure the modem at the other end of the link to answer calls.
– Set up the modem on the answering end to auto-answer.

Configuration Procedures

This section describes how to configure your router for V.25 bis. The tasks you need to perform are:

1. Adding V.25 bis addresses
2. Configuring V.25 bis parameters
3. Adding dial circuits
4. Configuring dial circuits

Note: You must restart the router for changes to the V.25 bis configuration to take effect.

Adding V.25 bis Addresses

You need to add a V.25 bis address for each local V.25 bis interface as well as for each destination. The V.25 bis address includes:

- *Address Name.* The address name is a description of the address. You can use any string of up to 23 printable ASCII characters.

- *Network Dial Address.* Telephone number of the local or destination port. You can enter up to 31 characters that are in the valid format of the connected V.25 bis modem.

  Note: The valid character set for telephone numbers as defined by the CCITT and supported by the IBM 2216 includes:

  – The decimal digits 0 through 9

  – Colon (:) – "Wait Tone"

  – Left-angled bracket (<) – "Pause", used for inserting a fixed delay (dependant on modem) between digit sequences. For example, when going through a PBX or PTN.

  – Equal (=) – "Separator 3", which is "for national use." (Consult your modem manual.)

  – The letter P – "Dialing to be continued in Pulse mode." (Not supported by some modems.)

  – The letter T – "Dialing to be continued in DTMF mode." (Not supported by some modems.)

To add a V.25 bis address, enter the **add v25-bis-address** command at the `Config>` prompt. For example:

```
Config>add v25-bis-address
Assign address name [1-23] chars []? remote-site-baltimore
Assign network dial address [1-20 digits][]? 1-909-555-1234
```

Configuring the V.25 bis Interface

This section explains how to configure the V.25 bis interface. To configure, do the following:

1. To set up a serial line interface for V.25 bis, set the data-link protocol for the serial line interface. From the `Config>` prompt, use the **set data-link v25bis** command. For example:

   ```
   Config>set data-link v25bis
   Interface Number [0]? 2
   ```

2. Display the `V.25bis Config>` prompt by entering the **network** command followed by the number of the interface. For example:

   ```
   Config>network 2
   V.25bis Data Link Configuration
   V25bis Config>
   ```

   You can use the **list devices** command at the `Config>` prompt to display a list of interface numbers configured on the router.

3. Use the **set local-address** command to specify the network address name of the local port. You must enter one of the address names you defined using the **add v25bis-address** command. For example:

   ```
   V25bis Config>set local-address
   Local network address name  []? remote-site-baltimore
   ```

Note: You must restart the router for configuration changes to take effect.

Optional V.25 bis Parameters

The following are optional V.25 bis parameters you can set. For a complete description of these commands, see "V.25 bis Configuration Commands" on page 35-5.

- You can limit the number of successive calls to an address that is inaccessible or that refuses those calls. To do so, use the **set retries-no-address** and the **set timeout-no-answer** commands.

- The **set disconnect-timeout** command controls the amount of time the router waits to initiate a call after dropping a signal from the previous call.

- The **set command-delay-timeout** command specifies the amount of time the router waits to initiate or answer a call after it turns on DTR.

- The **set connect-timeout** command specifies the number of seconds allowed for a call to be established.

- When you have finished configuring the interface, you can use the **list** command to display your configuration.

Adding Dial Circuits

Dial circuits are mapped to V.25 bis serial line interfaces. You can map multiple dial circuits to one serial line interface.

To add a dial circuit, use the **add device dial-circuit** command from the `Config>` prompt. The software assigns an interface number to each circuit. You will use this number to configure the dial circuit.

Example:

```
Config>add device dial-circuit
Adding device as interface 6
```

> **Note:** Dial circuits default to the Point-to-Point protocol (PPP). Although the set data-link command can be used to set the data-link of a dial circuit to Frame Relay, only PPP dial circuits are supported over V.25bis.

Configuring Dial Circuits

This section describes how to configure a dial circuit. For a complete description of the dial circuit commands, see Chapter 39, "Configuring Dial Circuits" on page 39-1. To configure the dial circuit, do the following:

1. Display the `Circuit Config>` prompt by entering the **network** command followed by the interface number of the dial circuit. You can use the **list devices** command at the `Config>` prompt to display a list of the dial circuits that you added. For example:

   ```
   Config>network 6
   Circuit configuration
   Circuit Config>
   ```

2. Map the dial circuit to a V.25 bis interface. The Base net is the V.25 bis interface number. For example:

   ```
   Circuit Config>set net
   Base net for this circuit [0]?  0
   ```

3. Specify the address name of the remote router to which the dial circuit will connect. You must use one of the names you defined using the **add v25-bis-address** command. For example:

   ```
   Circuit Config>set destination
   Assign destination address name []? newyork
   ```

4. Configure the dial circuit to initiate outbound calls only, accept inbound calls only, or both initiate and accept calls.

   Use the **set calls** command. To avoid a conflict if both ends of the link attempt to establish a call at the same time, configure the dial circuit at one end of the link to accept inbound calls only, and configure the dial circuit at the other end of the link to initiate outbound calls only. For example:

   ```
   Circuit Config>set calls outbound
   Circuit Config>set calls inbound
   ```

   > **Note:** For WAN-Restoral operations or another dial-on-demand application, you should set up the circuit for either inbound or outbound calls.

5. Specify the timeout period for the circuit.

   Use the **set idle** command. If there is no traffic over the circuit for this specified time period, the dial circuit hangs up. To configure the circuit as a dedicated circuit, set the idle timer to zero. To configure the circuit to dial on demand, set the idle timer to a value other than zero. The range is 0 to 65535 and the default is 60 seconds. For example:

   ```
   Circuit Config>set idle
   Idle timer (seconds, 0 means always active) [60]? 0
   ```

   > **Note:** For WAN-Restoral operations you must set the idle time to 0.

6. Optionally, you can delay the time between when a call is established and the initial packet is sent.

   Use the **set selftest-delay** command. Setting a selftest delay can prevent initial packets from being dropped. If your modems take extra time to synchronize, adjust this delay. For example:

```
Circuit Config>set selftest-delay
Selftest delay(milli-seconds,0 means no delay)[150]?200
```

7. Set the inbound address name.

   Use the **set inbound** command. You need to use this command only if you set up the circuit for both inbound and outbound calls and if the router's destination address is different from the destination address that the remote router dials. For example, the numbers would be different if one of the routers must go through a PBX, international, or inter-LATA exchange. For example:

   ```
   Circuit Config>set inbound
   Assign destination inbound address name []? newyork
   ```

   The inbound address name must match one of the names that you defined using the **add v25-bis-address** command.

8. Optionally, you can enter the configuration process for the data-link layer protocol that is running on the dial circuit (PPP or Frame Relay). Use the **encapsulator** command. For example:

   ```
   Circuit Config>encapsulator
   ```

V.25 bis Configuration Commands

Table 35-1 summarizes and the rest of the section explains the V.25 bis configuration commands. These commands allow you to display, create, or modify a V.25 bis configuration. Enter the V.25 bis configuration commands at the `V.25 bis Config>` prompt.

Table 35-1. V.25 bis Configuration Commands Summary

Command	Function
? (Help)	Lists the configuration commands or lists the options associated with that command.
List	Displays the V.25 bis configuration.
Set	Sets the local address, connect, disconnect, and no answer timeouts, number of retries after no answer, and command delay timeout.
Exit	Exits the V.25 bis configuration process and returns to the `Config>` prompt.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

```
LIST
SET
EXIT
```

Example: Set ?

V.25 bis Configuration Commands

```
COMMAND-DELAY-TIMEOUT
CONNECT-TIMEOUT
DISCONNECT-TIMEOUT
HDLC
LOCAL-ADDRESS
RETRIES-NO-ANSWER
TIMEOUT-NO-ANSWER
```

List

Use the **list** command to display the current V.25 bis configuration.

Syntax: list

Example: list

```
            V.25 bis Configuration

Local Network Address Name    = v403
Local Network Address         = 1-508-898-2403

Non-Responding addresses:
Retries            = 1
Timeout            = 0 seconds

Call timeouts:
Command Delay      = 0 ms
Connect            = 60 seconds
Disconnect         = 2 seconds

Cable type         = V.35 DTE
Speed              = 9600
```

Local Network Address Name:	Displays the network address name of the local port.
Local Network Address:	Displays the network dial address of the local port.
Non-responding addresses:	
Retries	Maximum number of calls the router attempts to make to a non-responding address during the timeout period.
Timeout	If the router reaches the maximum number of retries to a non-responding address, it does not attempt to establish the call until this time has expired. This timeout period begins when the router attempts the first call.
Call timeouts:	Number of call timeouts.
Command Delay	Amount of time, in milliseconds, that the router waits to initiate or answer a call after it turns on DTR (Data Terminal Ready). If you set this parameter to 0, the router waits for the modem to respond to DTR with the CTS (Clear to Send) signal before it issues commands.
Connect	Number of seconds allowed for a call to be established. If this parameter is set to 0, the modem controls the connection establishment timeout.
Disconnect	After the routers drops DTR it waits this amount of time before it initiates further calls. If you set this parameter to 0, the router waits for the modem to respond to the DTR drop by dropping CTS and DSR before it initiates the next call.

Set

Use the **set** command to configure local addresses, timeouts and delays for calls, retries and timeouts for non-responding addresses, and the HDLC cable type.

Syntax: <u>s</u>et   <u>com</u>mand-delay timeout . . .

              <u>con</u>nect-timeout . . .

              hdlc cable . . .

              hdlc speed . . .

              <u>d</u>isconnect-timeout . . .

              <u>l</u>ocal-address . . .

              <u>r</u>etries-no-answer . . .

              <u>t</u>imeout-no-answer . . .

command-delay-timeout *# of milliseconds*

After the router turns on DTR (Data Terminal Ready), it waits this amount of time before it initiates or answers a call. If you set this parameter to 0, the router waits for the modem to respond to DTR with the CTS (Clear to Send) signal before it issues commands. The range is 0 to 65535 milliseconds, and the default is 0.

Example: `set command-delay-timeout 0`

connect-timeout *# of seconds*

Sets the number of seconds allowed for a call to be established. The range is 0 to 65535 seconds, and the default is 60. If you set this parameter to 0, the modem controls the connection timeout. You should initially set this parameter to 0 and then use ELS event V25B.027 to find out how long it takes to establish connections to various destinations. You can then set this parameter to a number slightly higher than the longest connect time.

Note: Normally government regulation limits modem manufacturers to a maximum length for call setup. This value is merely an optimization, although inter-operation with some DSUs may require that you change this parameter.

Example: `set connect-timeout 10`

disconnect-timeout *# of seconds*

Specifies the amount of time, in seconds, that the router waits after dropping DTR before it initiates further calls. The range is 0 to 65535 seconds, and the default is 2. If you set this parameter to 0, the router waits for the modem to respond to the DTR drop by dropping CTS and DSR before it initiates the next call.

Example: `set disconnect-timeout 500`

hdlc cable *rs232 dte*

Specifies the type of cable connected to this interface. Setting this parameter allows you to view the cable type when you enter the **interface** command at the GWCON (+) prompt and when you enter the **statistics** command at the `V.25bis>` monitoring prompt. This parameter does not affect operation of the router.

Example: `set hdlc cable rs-232 dte`

hdlc speed

Specifies the line speed for this interface. Setting this parameter allows you to view the line speed when you enter the interface command at the GWCON (+) prompt and when you enter the statistics command at the `V.25bis>` monitoring prompt. The range is 2400 to 64 000 bps. The default is 9600 bps.

> **Note:** This command does not affect the actual line speed but it sets the speed some protocols, such as IPX, use when calculating routing cost parameters for dial circuits mapped to the V.25 bis interface.

Example: `set hdlc speed 2400`

local-address   *address name*

Specifies the network address name of the local port. This address name must match one of the names that you defined at the `Config>` using the **add v25-bis-address** command.

Example: `set local-address line-1-local`

retries-no-answer   *value*

Some telephone service providers impose restrictions on automatic recalling devices to limit the number of successive calls to an address that is inaccessible or that refuses those calls. This parameter specifies the maximum number of calls the router attempts to make to a non-responding address during the timeout period. The range is 0 to 10, and the default is 1.

> **Note:** Government regulation may also impose limits on the modem manufacturer that would supersede this parameter.

Example: `set retries-no-answer 2`

timeout-no-answer   *# of seconds*

After the router reaches the maximum number of **retries-no-answer** to a non-responding address, it does not initiate further calls to that address until this time has expired. This timeout period begins when the router attempts the first call to an address. The range is 0 to 65535 seconds, and the default is 0. If you set this parameter to 0, the modem controls the timeout period.

Example: `set timeout-no-answer 180`

Exit

Use the **exit** command to return to the `Config>` prompt.

Syntax: <u>e</u>xit

Example: `exit`

Chapter 36. Monitoring the V.25 bis Network Interface

This chapter describes the V.25 bis console commands and GWCON commands. It includes the following sections:

- "Accessing the Interface Console Process"
- "V.25 bis Console Commands"
- "V.25 bis and the GWCON Commands" on page 36-6

Accessing the Interface Console Process

To access the interface console process for V.25 bis, enter the following command at the GWCON (+) prompt:

    + **network** #

Where # is the number of the V.25 bis serial line. You cannot directly access the V.25 bis console process for dial circuits, but you can monitor the dial circuits that are mapped to the serial line interface.

Note: V.25 bis interfaces also have ELS troubleshooting messages that you can use to monitor V.25 bis related activity. See the *IBM Nways Event Logging System Messages Guide* for further details.

V.25 bis Console Commands

This section summarizes and explains the V.25 bis console commands. These commands allow you to view the calls, circuits, parameters, and statistics of the V.25 bis interfaces.

Enter the V.25 bis console commands at the V.25bis> prompt.

Table 36-1. V.25 bis Console Command Summary	
Console Command	**Function**
? (Help)	Lists the V.25 bis console commands or lists the options associated with specific commands.
Calls	List the number of completed and attempted connections made for each dial circuit mapped to this interface since the last time statistics were reset on the router.
Circuits	Shows the status of all data circuits configured on the V.25 bis interface.
Parameters	Displays the current parameters for the V.25 bis interface. (This command is similar to the V.25bis Config> list command.)
Statistics	Displays the current statistics for the V.25 bis interface.
Exit	Exits the V.25 bis console process and returns to the GWCON (+) process.

? (Help)

Use the **?** (help) command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: **?**

```
CALLS
CIRCUITS
PARAMETERS
STATISTICS
EXIT
```

Calls

Use the **calls** command to list the number of completed and attempted connections made for each dial circuit mapped to this interface since the last time statistics were reset on the router.

Syntax: <u>ca</u>lls

Example: **calls**

```
Net Interface  Site Name         In    Out   Rfsd  Blckd
1   PPP/0      v403              2     0     0     0

Unmapped connection indications:   0
```

Net	Number of the dial circuit mapped to this interface.
Interface	Type of interface and its instance number.
Site Name	Network address name of the dial circuit.
In	Number of inbound connections accepted for this dial circuit.
Out	Number of completed connections initiated by this dial circuit.
Rfsd	Number of connections initiated by this dial circuit that were refused by the network or the remote destination port.
Blckd	Number of connection attempts that the router blocked. The router blocks connection attempts if the local port is already in use, the maximum number of retries to a non-responding address is reached, or a modem is not responding.
Unmapped connection indications:	Number of connection attempts that were refused by the router because there were no enabled dial circuits that were configured to accept the incoming calls.

Circuits

The **circuits** command shows the status of all dial circuits configured on the V.25 bis port.

Syntax: <u>ci</u>rcuits

Example: **circuit**

```
Net Interface  MAC/Data-Link    State    Reason      Duration
2   PPP/0      Point to Point   Avail    Rmt Disc    1:02:25
```

Net	Number of the dial circuit mapped to this interface
Interface	Type of interface and its instance number.
MAC/DataLink	Type of datalink protocol configured for this dial circuit.
State	Current state of the dial circuit:
	Up - currently connected
	Available - not currently connected, but is available
	Disabled - dial circuit was disabled
	Down - failed to connect because of a busy dial circuit or because the link-layer protocol is down
Reason	Reason for the current state:
	nnn_Data - (where nnn is the name of a protocol) the circuit is Up because a protocol had data to send.
	Remote Disconnect - the circuit is either Down or Available because the remote destination disconnected the call.
	Operator Request - the circuit is Available because the last call was disconnected by a console command.
	Inbound - the circuit is Up because the circuit answered an inbound call.
	Restoral - the circuit is Up because of a WAN Restoral operation.
	Self Test - the circuit was configured as static (idle time=0) and successfully connected once it was enabled.
Duration	Length of time that the circuit has been in the current state.

Parameters

Use the **parameters** command to display the current V.25 bis serial line configuration. Note that this is the same information displayed in the V.25bis Config> list command.

Syntax: p̲arameters

Example: **parameters**

```
            V.25bis port Parameters

    Local Network Address Name   = v402
    Local Network Address        = 1-508-898-2402

    Non-Responding addresses:
    Retries          = 1
    Timeout          = 0 seconds

    Call timeouts:
    Command Delay    = 0 ms
    Connect          = 0 seconds
    Disconnect       = 0 seconds
```

Local Network Address Name:	Network address name of the local port.
Local Network Address:	Network dial address of the local port.

Non-responding addresses:	
Retries	Maximum number of calls the router attempts to make to a non-responding address during the timeout period.
Timeout	If the router reaches the maximum number of retries to a non-responding address, it does not attempt to establish the call until this time has expired. This timeout period begins when the router attempts the first call to an address.
Call timeouts:	
Command Delay	Amount of time, in milliseconds, that the router waits to initiate or answer a call after it turns on DTR (Data Terminal Ready). If you set this parameter to 0, the router waits for the modem to respond to DTR with the CTS (Clear to Send) signal before it issues commands.
Connect	Number of seconds allowed for a call to be established. If this parameter is set to 0, the modem controls the connection establishment timeout.
Disconnect	After the routers drops DTR it waits this amount of time before it initiates further calls. If you set this parameter to 0, the router waits for the modem to respond to the DTR drop by dropping CTS and DSR before it initiates the next call.

Statistics

Use the **statistics** command to display the current statistics for this V.25 bis interface.

Syntax: s̲tatistics

Example: **statistics**

```
                V.25bis port Statistics

      Adapter cable:               RS-232 DTE


       Nicknames:    RTS CTS DSR DTR DCD RI
       RS-232        CA  CB  CC  CD  CF  CE
       State:        OFF OFF OFF OFF OFF OFF

      Line speed:            4800
      Last port reset:       24 seconds ago

      Input frame errors:
       CRC error                   0    alignment (byte length)  0
       missed frame                0    too long (>  2182 bytes)  0
       aborted frame               0    DMA/FIFO overrun          0
       L & F bits not set          0
    Output frame counters:
       DMA/FIFO underrun errors    0    Output aborts sent        0
```

Adapter cable:	Type of adapter cable being used.
Nicknames:	Common names for the circuits.
RS-232	EIA 232 (also known as RS-232) names for the circuits.
State:	Current state of the circuits: ON, OFF, or "---," which means that the state is undefined for this type of interface.
Line speed:	The transmit clock speed (approximate).

Last port reset:	Length of time since the port was reset.
Input frame errors:	
CRC error	Number of packets received that contained checksum errors and as a result were discarded.
Alignment (byte length)	Number of packets received that were not an even multiple of 8 bits in length and as a result were discarded.
Missed Frame	When a frame arrives at the device and there is no buffer available, the hardware drops the frame and increments the missed frame counter.
too long (> nnnn bytes)	Number of packets received that were greater than the configured frame size (nnnn) and as a result were discarded.
aborted frame	Number of packets received that were aborted by the sender or a line error.
DMA/FIFO overrun	The number of times the serial interface card could not send data fast enough to the system packet buffer memory to receive packets from the network.
L & F bits not set	On serial interfaces, the hardware sets input-descriptor information for arriving frames. If the buffer can accept the complete frame upon arrival, the hardware sets both the last and first bits of the frame, indicating that the buffer accepted the complete frame. If either of the bits is not set, the packet is dropped, the L & F bits not set counter is incremented, and the buffer is cleared for reuse. **Note:** It is unlikely that the L & F bits not set counter will be affected by traffic.
Output frame counters:	
DMA/FIFO underrun errors	Number of times the serial interface card could not retrieve data fast enough from the system packet buffer memory to transmit packets onto the network.
Output aborts sent	Number of transmissions that were aborted as requested by upper-level software.

Exit

Use the **exit** command to return to the GWCON (+) prompt.

Syntax: e̲xit

Example: exit

V.25 bis and the GWCON Commands

While V.25 bis has its own console process for monitoring purposes, the router also displays configuration information and complete statistics for devices and circuits when you use the interface, statistics, and error commands from the GWCON environment. You can also use the GWCON **test** command to test DCEs and circuits.

Note: Issuing the **test** command to the V.25 bis serial interface causes the current call to be dropped and re-dialed.

For more information on the GWCON command, see Chapter 6, "The GWCON (Monitoring) Process and Commands" on page 6-1.

Statistics for V.25 bis Interfaces and Dial Circuits

Use the **interface** command at the GWCON (+) prompt to display statistics for V.25 bis serial line interfaces and dial circuits.

To display the following statistics for a V.25 bis serial line interface, use the **interface** command followed by the *interface number* of the V.25 bis serial line interface.

Example: `interface 10`

```
                                         Self-Test  Self-Test  Maintenance
       Nt Nt'  Interface  Slot-Port      Passed     Failed     Failed
       10 10   V.25/0     Slot: 4  Port: 0    1          0          0

         V.25bis Base Net MAC/data-link on EIA 232E/V.24 interface

         Adapter cable:            RS-232 DTE

         V.24 circuit: 105 106 107 108 109 125
         Nicknames:    RTS CTS DSR DTR DCD RI
         RS-232:       CA  CB  CC  CD  CF  CE
         State:        OFF OFF OFF ON  OFF OFF

         Line speed:           19.200 Kbps
         Last port reset:      55 minutes, 1 second ago

         Input frame errors:
          CRC error                     6    alignment (byte length)      0
          missed frame                  1    too long (> 2054 bytes)      0
          aborted frame                34    DMA/FIFO overrun             0
         Output frame counters:
          DMA/FIFO underrun errors      0    Output aborts sent           0
```

To display the following statistics for a dial circuit, use the **interface** command followed by the *interface number* of the dial circuit.

Example: `interface 29`

```
                            Self-Test  Self-Test  Maintenance
       Nt Nt'  Interface    Passed     Failed     Failed
       29 10   PPP/20          2          1          0

         Point to Point MAC/data-link on V.25bis Dial Circuit interface
```

The following table describes the output for both serial line interfaces and dial circuits.

Nt	Serial line interface number or dial circuit interface number.
Nt'	If "Nt" is a dial circuit, this is the interface number of the V.25 bis serial line interface to which the dial circuit is mapped.
Interface	Interface type and its instance number.
Slot	The slot number of the interface running V.25bis.
Port	The port number of the interface that is running V.25bis.
Self-Test Passed	Number of self-tests that succeeded.
Self-Test Failed	Number of self-tests that failed.
Maintenance: Failed	Number of maintenance failures.
Adapter cable:	Type of adapter cable that is being used.
V.24 circuit:	Circuit numbers as identified by V.24 specifications.
RS-232	EIA 232 (also known as RS-232) names for the circuits.
State	Current state of the circuits (ON or OFF).
Line speed	The transmit clock speed (approximate).
Last port reset	Length of time since the port was reset.
Input frame errors:	
CRC error	Number of packets received that contained checksum errors and as a result were discarded.
Alignment (byte length)	Number of packets received that were not an even multiple of 8 bits in length and as a result were discarded.
Missed Frame	When a frame arrives at the device and there is no buffer available, the hardware drops the frame and increments the missed frame counter.
too long (> nnnn bytes)	Number of packets received that were greater than the configured frame size and as a result were discarded.
DMA/FIFO overrun	The number of times the serial interface card could not send data fast enough to the system packet buffer memory to receive packets from the network.
L & F bits not set	On serial interfaces, the hardware sets input-descriptor information for arriving frames. If the buffer can accept the complete frame upon arrival, the hardware sets both the last and first bits of the frame, indicating that the buffer accepted the complete frame. If either of the bits is not set, the packet is dropped, the L & F bits not set counter is incremented, and the buffer is cleared for reuse.

Note: It is unlikely that the L & F bits not set counter will be affected by traffic. |
aborted frame	Number of packets received that were aborted by the sender or a line error.
Output frame counters:	
DMA/FIFO underrun errors	Number of times the serial interface card could not retrieve data fast enough from the system packet buffer memory to transmit packets onto the network.
Output aborts sent	Number of transmissions that were aborted as requested by upper-level software.

Chapter 37. Using and Configuring the ISDN Interface

This chapter describes the Integrated Services Digital Network (ISDN) interface on the IBM 2216. It includes the following sections:

- "ISDN Overview"
- "Sample ISDN Configurations" on page 37-4
- "Requirements and Restrictions for ISDN Interfaces" on page 37-6
- "Before You Begin" on page 37-6
- "Configuration Procedures" on page 37-7
- "ISDN Configuration Commands" on page 37-11.

ISDN Overview

The ISDN interface software allows you to interconnect routers over ISDN. You can set up the interface to act as a dedicated link or to initiate and accept switched-circuit connections, either on demand, automatically from restart, or on command by the operator.

ISDN Interfaces

The 2216 supports an ISDN Primary Rate Interface (PRI) that has an integrated CSU/DSU so that an external one is not required. The PRI provides up to 30 64-Kbps (kilobits per second) bearer (B) channels and one 64-Kbps data (D) channel. The B channels are used as HDLC frame delimited 64Kbps bit pipes. The D channel is used to set up calls.

- The PRI adapter does not support multipoint.
- The PRI adapter provides T1/J1 and E1 support.
    - T1/J1 supports 23 64Kbps B channels and one 64Kbps D channel.
    - E1 supports 30 64Kbps B channels and one 64Kbps D channel.
- The PRI adapter provides enhanced line ID (LID) support.

The ISDN interface establishes connections with a peer router over an ISDN connection. The interface accepts or initiates connections on command from dial circuits. Once the connection is established, the ISDN interface transparently passes data to and from the dial circuit.

Dial Circuits

Dial circuits are a set of potential point-to-point network connections that are mapped to ISDN interfaces. There are three types of dial circuits:

- Static circuits (or leased lines)
- Switched circuits that dial on demand and hang up after a specified idle time
- WAN restoral circuits that are used only when an assigned primary leased line fails

When bridging over a dial on demand interface it is recommended that you disable spanning tree for that interface and create MAC filters to filter out all undesired traffic. (The MAC filters would drop all frames that are not destined specific MAC addresses.) This keeps the dial circuit from staying connected due to unwanted traffic.

Note: You don't need to add any MAC filters when running BAN traffic on a FR dial-on-demand interface. The BAN software always performs filtering such that the

only bridging traffic that will keep a dial-on-demand circuit from hanging up is traffic whose destination MAC address matches the BAN DLCI MAC address.

Add a dial circuit for each potential destination. You can map multiple dial circuits to one ISDN interface. Each dial circuit is a normal serial line network, running Point-to-Point Protocol (PPP) or Frame Relay. These protocols are configured to operate over the dial circuits.

Note: You can assign a destination name to a **connection list** and assign a destination number to each line in the list. When that destination name is called, the numbers in the list are tried one by one until a connection is made or the list is exhausted.

Routing protocols cannot communicate directly with an ISDN interface. You need to configure these protocols to run on the dial circuits. This implementation supports the following protocols and features for ISDN dial circuits:

- APPN
- Banyan VINES
- DECnet
- DLSw
- IP
- IPX
- AppleTalk 2
- Bridging (SRB, STP, SR-TB, and SRT)
- Bandwidth reservation
- WAN restoral

Addressing

To place a telephone call, you need to specify the telephone number of the destination. To identify yourself to the destination, you need to specify your own telephone number. For ISDN, telephone numbers are called network dial addresses and, for convenience, they are given names called network address names that represent the telephone number.

When you set up an ISDN interface, you add addresses for each potential destination as well as for your own telephone number, which is called the local network address. When you configure a dial circuit, you set up the dial circuit with a local network address and one of the destination addresses.

Circuit Contention

An ISDN T1/J1 can have 23 active calls at a time and an ISDN E1 can have 30 active calls at one time. If the maximum number of dial circuits are active on the ISDN interface, other dial circuits configured for the same interface and the same priority cannot use it. The router drops packets sent by protocols on dial circuits that cannot connect to the ISDN destination.

See "Set" on page 39-3 for more information about priority.

Cost Control Over Demand Circuits

Dial-on-demand circuits always appear to be in the Up state to the routing protocols. Most protocols send out periodic routing information that could cause the router to dial out each time the routing information is sent over dial-on-demand circuits. To limit periodic routing updates, configure IP and OSI to use only static routes and disable the routing protocols (RIP, OSPF) over the dial circuits. If you are using IPX, configure it for low-frequency updates or change updates so that it sends broadcast messages at specified intervals, when there is data to send, or if there is a problem. You should also enable IPX Keepalive filtering, which prevents keepalive and serialization packets from continually activating a dial-on-demand link.

Call Verification

This ISDN implementation uses a proprietary caller-ID protocol to match incoming calls to dial circuits. The ID protocol uses the inbound and line ID name in the dial circuit configuration to match the dial circuit placing the call to the dial circuit that receiving the call. The caller-ID protocol is a brief identification protocol initiated by the caller and answered by the dial circuit receiving the call. If the caller does not provide the caller-ID message, the call may be rejected. The line ID exchanges occur on the B channel.

ISDN Cause Codes

This ISDN implementation specifies a cause code that will stop further processing to attempt to establish a connection through an ISDN interface. If the application retries, the router again attempts to establish a connection through this interface and will succeed if the original problem has been corrected. If during the retry the router encounters the same cause code, the router will not attempt further connection processing through this interface.

Sample ISDN Configurations

The following paragraphs show several typical ISDN configurations.

ISDN Connection with Four Routers

Figure 37-1 shows a sample ISDN configuration where each router has dedicated ISDN connections to two of its peers. Each link is bidirectional, forming a pair of counter-rotating rings, so that if any one link or router fails, none of the other routers is isolated.
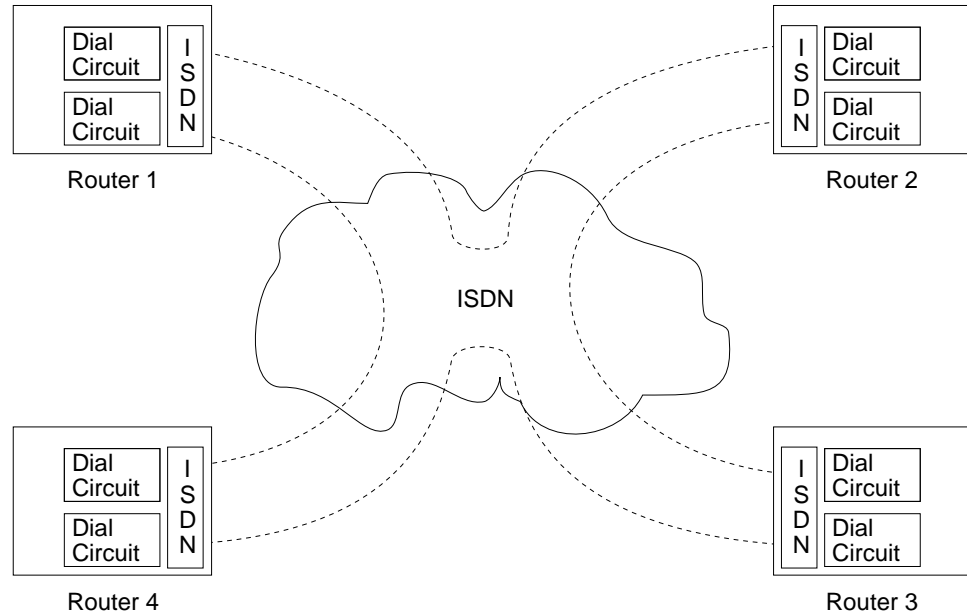
Figure 37-1. Sample ISDN Connection with Four Routers

Point-to-Point Configurations

In the ISDN configuration in Figure 37-2, both routers are in a point-to-point configuration, where there is one router on the ISDN line. In this case, the routers can use the B channels.
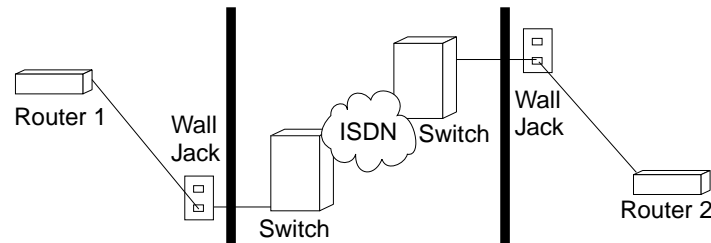
Figure 37-2. ISDN Point-to-Point Configuration

Frame Relay over ISDN Configuration

Figure 37-3 shows how you can connect to a Frame Relay network through an ISDN network. In this configuration, you set the data link on your dial circuits to Frame Relay.

Note: Dial circuits default to point-to-point (PPP) protocol. To change the protocol to Frame Relay, enter **set data-link fr** at the Config> prompt. A connection will only be usable if the data link on both ends matches (for example, either FR to FR, or PPP to PPP).
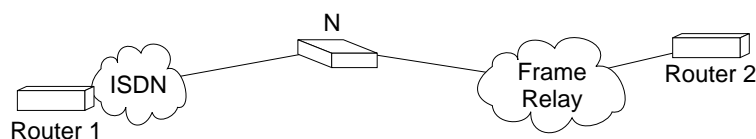
Figure 37-3. Frame Relay over ISDN Configuration

WAN Restoral Configuration

Figure 37-4 shows how you can use an ISDN connection to back up a failed dedicated WAN link (WAN restoral). In this example, Router A normally uses the WAN link to communicate with Router B. If that connection fails, the ISDN dial-up link reconnects the two routers. When the WAN link recovers, the secondary link automatically disconnects. For more information on how to configure the router for WAN restoral, see Chapter 14, "Configuring WAN Restoral" on page 14-1.
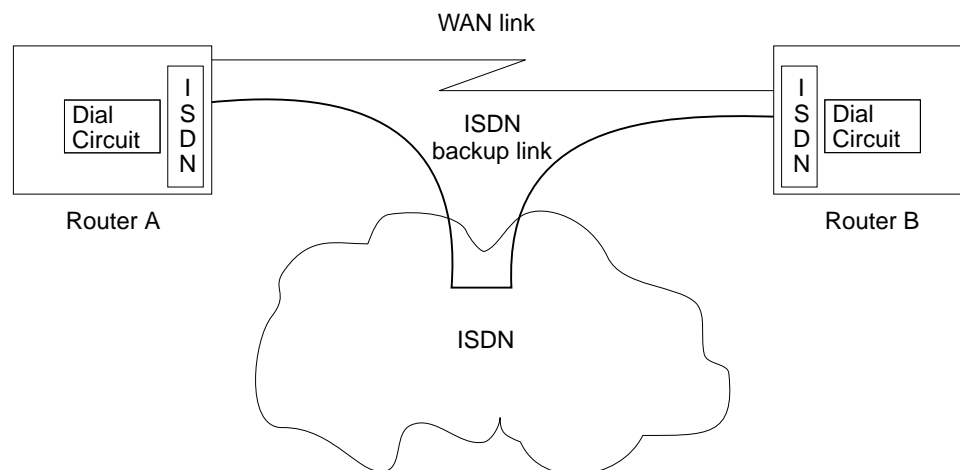
Figure 37-4. Using ISDN for WAN Restoral

For WAN Restoral, only dial circuits configured for PPP can be used as the secondary link. For WAN Reroute, either a PPP dial circuit or a FR dial circuit can be used as the alternate link.

Requirements and Restrictions for ISDN Interfaces

Switches Supported

The ISDN Primary Rate Interface (PRI) supports the following switches:

Switch names	Valid command
AT&T 5ESS (United States)	5ESS
AT&T 4ESS	4ESS
Australia (AUSTEL)	AUSPRI
INS-Net 1500 (Japan, NTT)	INSPRI
National ISDN 2	USNI2
NET 5 (Euro, ISDN, ETSI)	NET5
Northern Telecom 250 (DMS250)	DMS250

ISDN Interface Restrictions

- You cannot boot or dump the router over an ISDN interface.

- You cannot use the D channel for data traffic. The D channel is used only for setting up and taking down B channel connections.

- Optional ISDN network provider-supplied X.25 connectivity is not supported on the B channels.

- The option to multiplex two B channels to create a single, 128Kbps channel is not supported.

- The option to balance the traffic load on two B channels connected to the same destination, so that the total bandwidth is doubled is not supported.

Dial Circuit Configuration Requirements

You need to consider the following when you configure PPP or Frame Relay with ISDN:

- The ISDN interface will not enforce transmit delay counters that you set in the PPP configurations.

- Do not enable psuedo-serial-ethernet on the dial circuit.

Before You Begin

Before you configure ISDN, you need the following information:

- Telephone number of the local ISDN port.

- Destination telephone numbers, including any telephone extensions.

- Type of switch to which the ISDN interface is connected. See "Switches Supported" for the list of switches.

Note: TEI, SPID, and multipoint are not required for PRI.

Configuration Procedures

This section describes how to configure your ISDN interface and its associated dial circuits. Specifically, the tasks you need to perform are:

1. Adding ISDN addresses
2. Configuring ISDN parameters
3. Configuring the ISDN Interface
4. Adding dial circuits
5. Configuring dial circuits

Note: You must restart the router for configuration changes to take effect.

Adding ISDN Addresses

You need to add an ISDN address for each ISDN interface as well as for each destination. The ISDN address includes:

- *Address Name*. The address name is a description of the address. You can use any string of up to 23 printable ASCII characters.

- *Network Dial Address*. Telephone number of the local or destination port. You can enter up to 25 numbers as well as 6 characters, including punctuation. The router uses only the numbers.

- *Network Subdial Address*. Optional. This is an additional part of telephone number, such as an extension, that is interpreted once the interface connects to a PBX. You can enter up to 20 numbers, as well as 11 additional spaces and punctuation. The router uses only the numbers.

To add an ISDN address, enter the **add isdn-address** command at the Config> prompt. For example:

```
Config>add isdn-address
Assign address name [23] chars []? baltimore
Assign network dial address [1-15 digits] []? 1-555-0983
Assign network subdial address [1-20 digits] []? 23
```

To see a list of your ISDN addresses, enter **list isdn-address** at the Config> prompt.

To delete an ISDN address from your list, enter the **delete isdn-address** command at the Config> prompt.

Configuring ISDN Parameters

Access the ISDN Config> prompt. To access the ISDN Config> prompt, enter the **network** command followed by the interface number of the ISDN interface at the Config> prompt. For example:

```
Config>network 3
ISDN user configuration
ISDN Config>
```

You can use the **list devices** command at the Config> prompt to display a list of interface numbers configured on the router. See "ISDN Configuration Commands" on page 37-11 for more information about configuration commands.

1. Specify the type of switch to which this ISDN interface is connected.

Use the **set switch-variant** command to specify the type of switch to which this ISDN interface is connected. See "Switches Supported" on page 37-6 for the list of switches. For example:

```
ISDN Config>set switch net5
```

2. Specify the network address name of the local port.

Use the **set local-address-name** command to specify the network address name of the local port. You must use one of the address names you defined using the **add isdn-address** command. For example:

```
ISDN Config>: set local-address-name
Assign local address name []? baltimore
```

3. If you set the switch-variant to DMS250 or AUSPRI, set the directory number of the local port.

Enter the network dial address (telephone number) of the ISDN address that you entered using **set local-address-name** in 2.

```
ISDN Config>set dn0
Enter DN0 (Directory-Number-0) [ ]?15550983
```

Note: For PRI, only DN0 can be configured.

4. Set the frame size. Use the **set framesize** command. You must set the frame size so that it is greater than or equal to the frame size of the data-link protocol (FR) running on the link. For example:

```
ISDN Config>set framesize
Framesize in bytes (1024/2048/4096/8192) [1024]? 2048
```

Optional ISDN Parameters

This section describes optional ISDN parameters you can set. For a complete description of these commands see "ISDN Configuration Commands" on page 37-11.

* For all ISDN switches except INSPRI, you can limit the number of calls to an address that does not respond or that rejects those calls. Use the **set retries-call-address** command to set the number of calls to a non-responding destination. Use the **set timeout-call-address** command to set the time period to wait before trying the call again.

* If your ISDN telephone service provides accounting information, you can use the **add accounting entry** command to keep track of telephone charges.

When you have finished configuring the ISDN interface, you can use the **list** command to display your configuration.

Configuring the ISDN Interface

For the ISDN Primary Rate Interface (PRI) you need to configure either the T1/J1 or the E1 interface. You cannot configure both.

T1/J1 PRI Interface

Specify the following T1/J1 parameters:

1. For the T1/J1 PRI interface, line build out specifies the attenuation of the signal transmitted by the router's T1 port. Specify the `lbo` (line build out) based on the information provided by the service provider.

> a= -00.0 dB
> b= -07.5 dB

> c= -15.0 dB
> d= -22.5 dB

For example:

`set int lbo a`

2. Specify the `code`, either B8ZS or AMI. B8ZS is default. The service provider provides this information.

    For example:

    `set int code AMI`

3. Specify `ZBTSI`- Zero Byte Time Slot Inversion, either ENABLED or DISABLED. The default is DISABLED. The service provider provides this information.

    For example:

    `set int ZBTSI enabled`

4. Specify the `esf-data-link`. Select one of the following based on the service subscription:

    ANSI-T1.403   ANSI-IDLE   AT&T-IDLE

    Default is ANSI-T1.403

    For example:

    `set int esf-data-link ansi-idle`

E1 PRI Interface
For the E1 PRI interface, specify the following parameters:

1. Specify the `code`, either HDB3 or AMI. HDB3 is default. The service provider provides this information.

    For example:

    `set int code HDB3`

2. Specify the `crc4`, either ENABLED or DISABLED. Default is ENABLED. The service provider provides this information.

    For example:

    `set int crc4 enabled`

Adding Dial Circuits

Dial circuits are mapped to ISDN interfaces. You can map multiple dial circuits to one ISDN interface.

To add a dial circuit, enter the **add device dial-circuit** command at the `Config>` prompt. The software assigns an interface number to each circuit. You will use this number to configure the dial circuit. For example:

```
Config>add device dial-circuit
Adding device as interface 6
```

The number of dial circuits that can be configured depends on the total number of parameters to be configured and the size of the resulting configuration file up to a maximum of 236 dial circuits.

Note: Dial circuits default to point-to-point (PPP) protocol. To change the dial circuit protocol to Frame Relay, enter the **set data-link fr** command at the `Config>`

prompt. Other data-link types (X.25, V.25 bis, SDLC, and SRLY) are not supported.

Configuring Dial Circuits

This section describes how to configure a dial circuit.

1. Display the `Circuit Config>` prompt by entering the **network** command followed by the interface number of the dial circuit. You can enter the **list devices** command at the `Config>` prompt to display a list of the interface numbers configured on the router. For example:

   ```
   Config>network 6
   Circuit configuration
   Circuit Config>
   ```

2. Map the dial circuit to an ISDN interface. Use the **set net** command. The Base net is the ISDN interface number. For example:

   ```
   Circuit Config>set net
   Base net for this circuit [0]? 3
   ```

3. Specify the address name of the remote router to which the dial circuit will connect. You must use one of the names you defined using the **add isdn-address** command. For example:

   ```
   Circuit Config>set destination
   Assign destination address name []? baltimore
   ```

4. Configure the dial circuit to initiate outbound calls only, accept inbound calls only, or to both initiate and accept calls.

   Use the **set calls** command. To avoid a conflict if both ends of the link attempt to establish a call at the same time, configure the dial circuit at one end of the link to accept inbound calls only, and configure the dial circuit at the other end of the link to initiate outbound calls only. For example:

   ```
   Circuit Config>set calls outbound
   Circuit Config>set calls inbound
   Circuit Config>set calls both
   ```

   Note: For WAN-Restoral operations or another dial-on-demand application, you should set up the circuit for either inbound or outbound calls.

5. Specify the timeout period for the circuit.

   Use the **set idle** command. If there is no traffic over the circuit for this specified time period, the dial circuit hangs up. To configure the circuit as a dedicated circuit, set the idle timer to zero. To configure the circuit to dial on demand, set the idle timer to a value other than zero. The range is 0 to 65535 and the default is 60 seconds. For example:

   ```
   Circuit Config>set idle
   Idle timer (seconds, 0 means always active) [0]? 0
   ```

   Note: For WAN-Restoral operations, you must set the idle timeout to 0. (Use the **set idle** command from the `Circuit Config>` prompt. See Chapter 39, "Configuring Dial Circuits" on page 39-1.)

6. Optionally, you can provide a name for a dial circuit by specifying a lid_out_addr.

   When more than one circuit is configured between two routers (parallel circuits), there must be a way to know which dial circuit connects them. For this purpose, a lid_out_addr is sent from the router at one end (the caller). At the receiving end the other router configures the same string as the inbound

destination name. The lid_out_addr must be an address name that has been previously added using "ADD ISDN-ADDRESS" at the **config>** prompt.

```
Circuit Config>set lid_out_addr  router2
```

7. Optionally, you can set the relative priority of dial circuits.

The priority field allows a circuit to preempt another when no channels are available. If a call request is made and all the channels are in use, then the priority of the requesting dial circuit is checked against all the active dial circuits. If there is one whose priority is lower than this, then that circuit is disconnected and a call is made for the higher priority dial circuit.

See "Set" on page 39-3 for more information about priority.

```
Circuit Config>set priority 1
```

8. Optionally, you can delay the time between when a call is established and the initial packet is sent. Use the **set selftest-delay** command. Some ISDN switches start to send data before receiving a signal indicating the complete establishment of the circuit at the destination. Setting a selftest delay can prevent initial packets from being dropped. For example:

```
Circuit Config>set selftest-delay
Selftest delay(milli-seconds,0 means no delay)[150]?200
```

9. Set the inbound address name.

Use the **set inbound** command. You need to use this command only if you set up the circuit for both inbound and outbound calls and if the router's destination address is different from the destination address that the remote router dials. For example, the numbers would be different if one of the routers must go through a PBX, international, or inter-LATA exchange. For example:

```
Circuit Config> set inbound
Assign destination inbound address name [ ]? newyork
```

The inbound address name must match one of the names you defined using the **add isdn-address** command.

10. Optionally, you can enter the configuration process for the data-link layer protocol that is running on the dial circuit (PPP or Frame Relay).

Use the **encapsulator** command. For example:

```
Circuit Config> encapsulator
```

ISDN Configuration Commands

Table 37-1 on page 37-12 summarizes the ISDN configuration commands, and the following sections explain the commands. Enter these commands at the ISDN Config> prompt.

Table 37-1. ISDN Configuration Command Summary	
Command	**Function**
? **(Help)**	Lists the configuration commands or lists the options associated with that command.
Add	Adds accounting entries to the ISDN configuration.
Disable	Valid only for BRI. Disables Power Source 1 detection.
Enable	Valid only for BRI. Enables Power Source 1 detection.
List	Displays the ISDN configuration.
Remove	Removes accounting entries from the ISDN configuration.
Set	Sets the frame size, local address, no-answer timeouts, number of retries after no answer, type of ISDN switch, directory numbers, and TEI.
Cause Codes	Stops futher processing attempts to establish a connection through an interface.
Exit	Exits the ISDN configuration process and returns to the Config> prompt.

? (Help)

The **?** (help) command lists available commands. You can also enter ? after a command to list its options.

Syntax: ?

Example: `set ?`

```
FRAMESIZE
LOCAL-ADDRESS-NAME
RETRIES-CALL-ADDRESS
TIMEOUT-CALL-ADDRESS
SWITCH-VARIANT
DN0 (Directory Number 0)
INTERFACE
```

Add

The **add** command lets you add accounting entries to your ISDN configuration.

Syntax: add   <u>acc</u>ounting-entry ...

accounting entry   *name*
> If your ISDN telephone service provides accounting information, you can use accounting entries to track telephone charges accrued for specified network addresses. You can add up to eight entries for each ISDN interface. The accounting entry name must match one of the ISDN addresses you added using the **add isdn-address** command at the Config> prompt.

> **Example:** `add accounting-entry`
>
> > Assign accounting entry name []? **baltimore**

To display accrued telephone charges, enter the accounting command at the ISDN console prompt.

List

The **list** command displays the current ISDN configuration.

Syntax: list

Example: `list`

```
                          ISDN Configuration

Local Network Address Name      = local2216
Local Network Address:Subaddress = 2542216:

Maximum frame size in bytes     = 2048
Outbound call address Timeout   =    0  Retries =   0
Switch Variant                  = NT DMS-250
DN0 (Directory Number 0)        = 2542216
No circuit address accounting information being kept

T1/J1 Interface Parameters:

LBO           = 00.0 dB
Code          = B8ZS
ZBTSI         = Disabled
ESF-Data-Link = ANSI-IDLE
```

Remove

The **remove** command lets you remove accounting entries you set using the **add accounting-entry** command.

Syntax: remove   accounting-entry...

Example: `remove accounting-entry`

```
          Remove accounting entry name []? baltimore
```

Set

The **set** command configures frame size, addresses, and timeouts. It also specifies the switch-variant and TEI number. For PRI, the terminal endpoint identifier (TEI) is always zero (0).

Syntax: set   framesize ...
                         interface
                         local-address-name...
                         retries-call-address...
                         switch-variant...
                         dn0...

framesize   *1024* or *2048* or *4096* or *8192*
   Sets the size of the network layer portion of frames transmitted and received on the ISDN interface. Data link and MAC layer headers are not included. You must set the frame size so that it is greater than or equal to the frame size of the data-link protocol (PPP) running on the link.

   For PPP, you can change the frame size using the **set hdlc frame-size** command at the PPP Config> prompt.

   Example: `set framesize`

```
          Framesize in bytes (1024/2048/4096/8192) [1024]? 2048
```

interface
> For PRI only. Sets the following interface parameter values for T1 and E1 lines.

For T1 PRI:

lbo
> The attenuation of the signal transmitted by the router's T1 port. This information is provided by the service provider.
>
> **Valid Values**:
>
> > a= -00.0 dB
> > b= -07.5 dB
> > c= -15.0 dB
> > d= -22.5 dB
>
> **Default Value**: a

code
> This information is provided by the service provider.
>
> **Valid Values**: B8ZS or AMI
>
> **Default Values**: B8ZS

ZBTSI
> Zero Byte Time Slot Inversion. This information is provided by the service provider.
>
> **Valid Values**: Enabled or Disabled
>
> **Default Value**: Disabled

esf-data-link
> The service subscription. This information is provided by the service provider.
>
> **Valid Values**:
>
> > ANSI-T1.403
> > ANSI-IDLE
> > AT&T-IDLE
>
> **Default Value**: ANSI-T1.403

For E1 PRI:

code
> This information is provided by the service provider.
>
> **Valid Values**: HDB3 or AMI
>
> **Default Value**: HDB3

crc4
> Specifies whether the router's E1 port will transmit crc4 code words and check them in the received frames. this information is provided by the service provider.
>
> **Valid Values**: Enabled or Disabled
>
> **Default Value**: Enabled

local-address-name   *address name*
> This is the network address name of the local ISDN interface. This address name must match one of the names that you defined at the `Config>` prompt using the **add isdn-address** command.
>
> **Valid Values**: Any valid address
>
> **Default Value**: None
>
> **Example: `set local-address-name`**

```
              Assign local address name []? line-1-local
```

retries-call-address *value*

Some telephone service providers impose restrictions on automatic recalling devices to limit the number of successive calls to an address that is inaccessible or that refuses those calls. **Retries-call-address** specifies the maximum number of calls the router attempts to make to a non-responding address during the timeout period. Setting **retries-call-address** to 0 causes the router to retry until the call is established.

If you set the switch-variant to INS64, you cannot change **retries-call-address**. It is fixed at 2.

Valid Values: 0 to 10

Default Value: 2

Example: set retries-call-address

```
         Outbound call address retries [0]? 2
```

switch-variant *4ess* or *5ess* or *auspri usni2* or *ins1500* or *dms250*

Specifies the model of the switch to which this ISDN interface is connected. The default for the ISDN Primary Rate interface is DMS250. You can choose switch-variants for the ISDN Primary Rate interface from the following lists.

Valid Values ISDN Primary Rate Interface (PRI):

- AT&T 5ESS (United States)
- AT&T 4ESS
- Australia (AUSTEL)
- INS-Pri (Japan, NTT)
- National ISDN 2
- NET 5 (Euro, ISDN, ETSI)
- Northern Telecom 250

Default Value: DMS250

Example: set switch-variant

```
         Switch-Variant-Model []? net5
```

dn0 *directory number 0*

If you set the switch-variant to AUSPRI, set the directory number of the local port. **Dn0** must match the network dial address (telephone number) you configured using the **set local-address-name** command.

Example: set dn0

```
         Enter DN0 (Directory-Number-0) [ ]? 15088981234
```

Cause Codes

Use the **Cause Code** command to prevent the router from retrying to establish a connection through the interface when it receives a "user busy" response.

Valid Values: X'01' to X'FE'

Default Value: None

Syntax: <u>ca</u>use code code

Example: `cause code X'0F'`

See *Event Logging System Messages Guide* for a complete list of ISDN cause codes.

Exit

Use the **exit** command to return to the `Config>` prompt.

Syntax: <u>e</u>xit

Example: `exit`

Chapter 38. Monitoring the ISDN Interface

This chapter describes the ISDN console commands and GWCON commands. It includes the following sections:

- "Accessing the Interface Console Process"
- "ISDN Console Commands"
- "ISDN and the GWCON Commands" on page 38-6

Note: ISDN interfaces also have ELS messages and cause codes that you can use to monitor ISDN-related activity. See *Event Logging System Messages Guide*.

Accessing the Interface Console Process

To access the interface console process for ISDN, enter the following command at the GWCON (+) prompt:

> + **network** #

Where # is the number of the ISDN interface. You cannot directly access the console process for dial circuits, but you can monitor the dial circuits that are mapped to the ISDN interface.

ISDN Console Commands

The following sections explain the ISDN console commands which allow you to view the accounting entries, calls, circuits, parameters, and statistics of the ISDN interfaces. Enter these commands at the ISDN> prompt.

Table 38-1. ISDN Console Command Summary

Console Command	Function
? (Help)	Lists the ISDN console commands or lists the options associated with specific commands.
Accounting	Displays the telephone charges that have accrued for each network address configured by the **add accounting entries** ISDN configuration command.
Calls	List the number of completed and attempted connections made for each dial circuit mapped to this interface since the last time statistics were reset on the router.
Channels	Lists the statistics for the channels on the ISDN Primary Rate Interface.
Circuits	Shows the status of all data circuits configured on the ISDN interface.
Parameters	Displays the current parameters for the ISDN interface.
Statistics	Displays the current statistics for the ISDN interface.
Exit	Exits the ISDN console process and returns to the GWCON (+) process.

? (Help)

The **? (help)** command lists available commands. You can also enter ? after a command to list its options.

Syntax: ?

Example: ?

```
ACCOUNTING
CALLS
CIRCUITS
PARAMETERS
STATISTICS
CAUSE_CODE
CHANNELS
EXIT
```

Accounting

If you set up accounting entries using the **add accounting entries** ISDN command, the **accounting** command displays accrued telephone charges for each network address that you added.

Syntax: accounting

Example: accounting

```
Address Name    Address      SubAddress          Charge
vl2-31          21           1                   0.0
vl2-33          20           1                   0.0
vl_2-31         021          001                 0.0
vl_2-33         020          001                 0.0
ydc100          100                              0.0
ydc200          200                              0.0

All others:                                      0.0
```

Calls

Use the **calls** command to list the number of completed and attempted connections made for each dial circuit mapped to this interface since the last time statistics were reset on the router.

Syntax: calls

Example: calls

```
Net Interface  Site Name        In    Out   Rfsd  Blckd
  4     PPP/1  v403              2     0     0     0

Unmapped connection indications:   0
```

Net	Number of the dial circuit mapped to this interface.
Interface	Type of interface and its instance number.
Site Name	Network address name of the dial circuit.
In	Inbound connections accepted for this dial circuit.
Out	Completed connections initiated by this dial circuit.
Rfsd	Connections initiated by this dial circuit that were refused by the network or the remote destination port.

Blckd	Connection attempts that the router blocked. The router blocks connection attempts if the local port is already in use, the maximum number of retries to a non-responding address is reached, or a modem is not responding.
Unmapped connection indications:	
	Connection attempts that were refused by the router because there were no enabled dial circuits that were configured to accept the incoming calls.

Channels

The **channels** command lists the statistics for a channel on the ISDN Primary Rate Interface.

Circuits

The **circuits** command shows the status of the dial circuits configured on the ISDN interface that are in the state of "Up" or "Available."

Syntax: <u>ci</u>rcuits

Example: `circuit`

```
Net Interface MAC/Data-Link   State    Reason     Duration
4    PPP/1    Point to Point  Up B1    SelfTest   91:24:03
5    PPP/2    Point to Point  Up B2    Inbound    91:24:00
```

Net	Number of the dial circuit mapped to this interface
Interface	Type of interface and its instance number.
MAC/Data-Link	Type of data-link protocol configured for this dial circuit.
State	Current state of the dial circuit:

	Up	Currently connected.
	Available	Not currently connected, but available.
	Disabled	Dial circuit disabled.
	Down	Failed to connect because of a busy dial circuit or because the link-layer protocol is down.

Reason	Reason for the current state:

	nnn_Data	(Where nnn is the name of a protocol.) The circuit is up because a protocol had data to send.
	Rmt Disc	Remote Disconnect. The circuit is either down or available because the remote destination disconnected the call.
	Opr Req	Operator Request. The circuit is available because the last call was disconnected by a console command.
	Inbound	The circuit is up because the circuit answered an inbound call.

	Restoral	The circuit is up because of a WAN-Restoral operation.
	Self Test	The circuit was configured as static (idle time=0) and successfully connected once it was enabled.
Duration		Length of time that the circuit has been in the current state.

Parameters

Use the **parameters** command to display the current ISDN configuration.

Syntax: parameters

Example: parameters

```
ISDN Port parameters:

Local Address Name:      vl233
Local Network Address:   20
Local Network Subaddress:
Frame Size:        2048
TEI for B-CH 1:    Automatic
TEI for B-CH 2:    Automatic
Switch Variant:    AT&T 5ESS (United States)
Multipoint Selection:   Multipoint
Directory Number 0:     20
Outbound call address Timeout:    180     Retries:    0

Accounting Name         Network Address   Network Subaddress
---------------         ---------------   ------------------
vl215                   22
vl218                   22
vl231                   21
vl233                   20
```

Statistics

Use the **statistics** command to display the current statistics for this ISDN interface.

Syntax: statistics

Example for PRI with E1: statistics

```
     Link:   Active   ISDN Firmware:   1.0   Handler State: Running

     Transmit   D  Channel     Receive   D  Channel

     Packets        68422      Packets        68419
     Bytes         411656      Bytes         413592
     Overflow          23      Overflow           3
     Underrun           0      Too Long           6
                               Abort              4
                               CRC error          8
                               Misaligned         3

     Transmit   B  Channels    Receive   B  Channels

     Packets      1499094      Packets      1499228
     Bytes       59955660      Bytes       59951780
     Overflow           0      Overflow          90
     Underrun           0      Too Long         171
                               Abort            139
                               CRC error        232
                               Misaligned        72

     E1 Status Register        E1 Error Count Registers

     Receive AIS       : Off  CRC6 Errors:        4
     Receive RAI       : Off  LCV  Errors:       38
```

```
                          Receive Carrier Loss: Off FEB  Errors:        11
                          Receive Loss of Sync: Off FAS  Errors:        24
```

Example for PRI with T1: `statistics`

```
        Transmit    D  Channel      Receive    D  Channel

        Packets              0      Packets              0
        Bytes                0      Bytes                0
        Overflow         68480      Overflow             0
        Underrun             0      Too Long             0
                                    Abort                0
                                    CRC error            0
                                    Misaligned           0

        Transmit    B  Channels     Receive    B  Channels

        Packets              0      Packets              0
        Bytes                0      Bytes                0
        Overflow             0      Overflow             9
        Underrun             0      Too Long             0
                                    Abort                0
                                    CRC error            0
                                    Misaligned           0

        T1 Status Register          T1 Error Count Registers

        Receive AIS        : Off    LCV  Errors:          0
        Receive RAI        : Off    CRC6 Errors:          0
        Receive Carrier Loss: Off   Sync Errors:   47937328
        Receive Loss of Sync: On

        T1 PRM Events                   Local        Remote

        CRC Error                         0             0
        Controlled Slip                   0             0
        Line Code Violation               0             0
        Frame Sync Bit Error              0             0
        Severely Errored Frame            0             0
        Payload Looback Active            0             0
        PRMs Processed (1/sec)            0             0
```

Exit

Use the **exit** command to return to the GWCON (+) prompt.

Syntax: ḙxit

Example: `exit`

ISDN and the GWCON Commands

While ISDN has its own console process for monitoring purposes, the router also displays configuration information and complete statistics for devices and circuits when you use the **interface, statistics,** and **error** commands from the GWCON environment. You can also use the GWCON **test** command to test DCEs and circuits.

Note: Issuing the **test** command to the ISDN interface causes the current call to be dropped and re-dialed.

Interface — Statistics for ISDN Interfaces and Dial Circuits

Use the **interface** command at the GWCON prompt (+) to display statistics for ISDN interfaces and dial circuits.

To display statistics for a dial circuit, enter the **interface** command followed by the interface number of the dial circuit. For ISDN interfaces, information is displayed on a D and B channel basis. (This is the same information that is displayed by the ISDN **statistics** command.)

Example: `interface 2`

```
                                   Self-Test  Self-Test  Maintenance
        Nt Nt'  Interface  Slot-Port        Passed     Failed     Failed
        2  2    ISDN/0     Slot: 8 Port: 1     1          0          0

          ISDN Base Net MAC/data-link on ISDN Primary Rate interface
          Link:    Active    ISDN Firmware:  1.0    Handler State: Running

        Transmit    D  Channel     Receive    D  Channel

        Packets          36      Packets          36
        Bytes           214      Bytes           214
        Overflow          0      Overflow          0
        Underrun          0      Too Long          0
                                 Abort             0
                                 CRC error         0
                                 Misaligned        0

        Transmit    B  Channels    Receive    B  Channels

        Packets           0      Packets           0
        Bytes             0      Bytes             0
        Overflow          0      Overflow          0
        Underrun          0      Too Long          0
                                 Abort             0
                                 CRC error         0
                                 Misaligned        0

        T1 Status Register       T1 Error Count Registers

        Receive AIS        : Off  LCV  Errors:          0
        Receive RAI        : Off  CRC6 Errors:          0
        Receive Carrier Loss: Off  Sync Errors:          0
        Receive Loss of Sync: Off

        T1 PRM Events              Local      Remote

        CRC Error                    0          0
        Controlled Slip              0          0
        Line Code Violation          0          0
        Frame Sync Bit Error         0          0
        Severely Errored Frame       0          0
        Payload Looback Active       0          0
        PRMs Processed (1/sec)     365        367
```

To display the following statistics for a dial circuit, use the **interface** command followed by the interface number of the dial circuit.

Example: `interface 3`

```
                                      Self-Test  Self-Test  Maintenance
              Nt Nt'  Interface       Passed     Failed     Failed
              3  2    PPP/1           1          0          0

              Point to Point MAC/data-link on ISDN Primary Rate interface
```

The following list describes the output for both ISDN and dial circuits.

Nt	Serial line interface number or dial circuit interface number.
Nt'	If *Nt* is a dial circuit, this is the interface number of the ISDN interface to which the dial circuit is mapped.
Interface	Interface type and its instance number.
Slot	The slot that contains the ISDN adapter
Port	The port number on the ISDN adapter.
Self-Test Passed	Number of self-tests that succeeded.
Self-Test Failed	Number of self-tests that failed.
Maintenance: Failed	Number of maintenance failures.

Configuration — Information on Router Hardware and Software

Enter the **configuration** command at the GWCON (+) prompt to display information about the router hardware and software. It includes a section that displays the interfaces configured on the router along with the state of the interface.

If a dial circuit is configured to dial-on-demand, the state of the dial circuit is always displayed as Up whether or not it is connected. In this case Up means that the dial circuit is either connected or available.

If a dial circuit is configured as a static circuit, the state indicates Up only if the dial circuit is connected. (Refer to "Configuration" on page 6-4 for a sample output from the **configuration** command.

Chapter 39. Configuring Dial Circuits

This chapter describes the dial circuit configuration commands that are relevant to a PPP dial circuit interface mapped to a V.25bis or ISDN interface and a Frame Relay dial circuit interface mapped to an ISDN interface.

For information on how to configure dial circuits for use with:

- ISDN interfaces, see Chapter 37, "Using and Configuring the ISDN Interface" on page 37-1.
- V.25 bis interfaces, see Chapter 35, "Using and Configuring the V.25 bis Network Interface" on page 35-1.

Dial Circuit Configuration Commands

This section summarizes and explains the dial circuit configuration commands.

Table 39-1 summarizes the dial circuit configuration commands. Enter the dial circuit configuration commands at the `Circuit Config>` prompt. You must restart the router for configuration changes to take effect.

To access the `Circuit Config>` prompt, enter the **network** command followed by the interface number of the "dial circuit." (The dial circuit number was assigned when you entered the **add device dial-circuit** command.) You can enter the **list devices** command at the `Config>` prompt to display a list of the dial circuits that you added.

Table 39-1. Dial Circuit Configuration Commands Summary	
Command	**Function**
? (Help)	Lists the configuration commands or lists the options associated with that command.
Delete	Deletes the inbound call settings from the dial circuit configuration.
Encapsulator	Allows you to change the data-link protocol configuration.
List	Displays the dial circuit configuration parameters.
Set	Configures the dial circuit for inbound or outbound calls, maps the dial circuit to a serial line interface, and sets addresses, idle timeout, priority, and self-test delay.
Exit	Exits the dial circuit configuration process and returns to the `Config>` prompt.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a **?** after a specific command name to list its options.

Syntax: ?

Example: ?

```
DELETE
ENCAPSULATOR
LIST
SET
EXIT
```

Example: Set ?

```
NET
CALLS
DESTINATION
INBOUND DESTINATION
ANY_INBOUND
IDLE
LID_OUT_ADDR
PRIORITY
SELFTEST-DELAY
```

Delete

Use the **delete** command to remove the inbound call settings from the dial circuit configuration.

Syntax: delete inbound destination

inbound destination
> Removes both the INBOUND destination and the ANY_INBOUND settings from the dial circuit configuration. This causes the dial circuit to accept calls only from callers that have a phone number that matches the DESTINATION parameter.

Example: delete inbound

Encapsulator

Use the **encapsulator** command to enter the configuration process for the link-layer protocol that is running on the dial circuit. The default protocol for dial circuits is PPP. For ISDN dial circuits, you can change the protocol to frame relay by using the **set data-link fr** command at the Config> prompt.

Syntax: encapsulator

Example: encapsulator

```
Point-to-Point user configuration
PPP Config>
```

Be aware of the following when you configure PPP for V.25 bis:

- The V.25 bis interface predefines clocking as external and encoding as NRZ. The modem (DCE) controls the clock speed. You cannot configure clocking, encoding, and other HDLC parameters as part of the PPP dial circuit configuration.

Be aware that you cannot configure HDLC parameters of the dial circuit configuration when you configure PPP or Frame Relay for ISDN. Physical layer parameters are configured on the ISDN interface.

For information on configuring the PPP protocol, refer to Chapter 23, "Configuring Serial Line Interfaces" or refer to Chapter 28, "Using and Configuring Point-to-Point Protocol Interfaces."

For information on configuring the Frame Relay protocol, refer to Chapter 26, "Using and Configuring Frame Relay Interfaces" or refer to Chapter 27, "Monitoring Frame Relay Interfaces."

To return to the `Circuit Config>` prompt, use the **exit** command.

List

Use the **list** command to display the current dial circuit configuration.

Syntax: l̲ist

Example: l̲i̲s̲t̲

```
        Base net:         1
        Destination name:    remote-site-baltimore
        Inbound dst name:    local-1
        Outbound calls       allowed
        Inbound calls        allowed
        Idle timer         = 60 sec
        SelfTest Delay Timer = 0 ms
```

Base net:	Name of the serial line interface to which this dial circuit is mapped.
Destination name:	Network address name to be called for outbound circuits, and the default comparison address used by the caller-ID mechanism for inbound calls.
Inbound dst name:	This parameter appears only if the circuit is configured to accept inbound calls that do not match any other addresses.
Inbound dst name:	Alternate comparison address name used by the caller-ID mechanism for inbound calls.
Outbound calls allowed	Displays this parameter when the circuit is configured to initiate outbound calls.
Inbound calls allowed	Displays this parameter when the circuit is configured to accept inbound calls.
Idle timer	Displays the idle timer setting in seconds. The range is 0 to 65535; 0 indicates that this is a dedicated circuit (leased line).
SelfTest Delay Timer	Displays the self-test delay timer setting in milliseconds. The range is 0 to 65535; 0 indicates no delay.

Set

Use the **set** command to map the dial circuit to an interface (eg. ISDN or V.25 bis), configure the dial circuit for inbound and/or outbound calls, and set destination addresses, inbound addresses, idle timeout, and self-test delay.

Syntax: s̲et   n̲et...
          c̲alls...
          d̲estination...
          i̲n̲bound destination...
          a̲n̲y_inbound
          i̲dle...
          l̲id_out_addr...
          p̲riority...
          s̲elftest-delay...

net *# of serial line interface*
>   Specifies the number of the ISDN or V.25 bis serial line interface to which you
>   want to map this circuit.

>   **Example:**
>
>    ```
>    Circuit Config>set net
>    Base net for this circuit [0]? 2
>    ```

calls *outbound* or *inbound* or *both*
>   Restricts this dial circuit to initiating outbound calls only, accepting inbound
>   calls only, or both initiating and accepting calls. The default is both.

>   **Note:** If you are using this circuit for WAN-Restoral or another dial-on-demand
>   application, you should set the calls to either **inbound** or **outbound**. This
>   avoids a conflict if both ends of the circuit attempt to establish a call at the
>   same time.

>   **Example:**   `set calls outbound`

destination *address name*
>   This parameter is required for the dial circuit to operate. It specifies the
>   network dial address of the remote router to which this dial circuit will connect.
>   The caller-ID protocol uses this parameter as the default comparison address
>   for incoming calls. This parameter must match an address name that you
>   assigned using the `Config>` prompt using either the **add isdn address**
>   command or the **add v25-bis address** command.

>   **Example:**   `set destination remote-site-baltimore`

inbound destination *address name*
>   Set this parameter if the dial circuit is set up for both inbound and outbound
>   calls and if this router's local dial address is different from the destination dial
>   address that the remote router dials. For example, the numbers would be
>   different if one of the routers must go through a PBX, international, or
>   inter-LATA exchange. This parameter overrides the default comparison
>   address that the caller-ID protocol uses for incoming calls. This parameter
>   must match an address name that you assigned at the `Config>` prompt using
>   either the **add isdn address** command or the **add v25-bis address** command.

>   **Example:**   `set inbound remote-site-1`

any_inbound
>   Specifies that inbound calls that do not match any other dial circuit will be
>   mapped to this circuit and accepted as inbound calls.

>   **Example:**   `set any_inbound`

idle *# of seconds*
>   Specifies a timeout period for the circuit. If there is no protocol traffic over the
>   circuit for this specified time period, the dial circuit hangs up. The range is 0 to
>   65535, and the default is 60 seconds. A setting of zero specifies that there is
>   no timeout period and that this is a dedicated circuit (leased line).

>   **Note:** For WAN Restoral operations, you must set the idle timeout to 0.

>   **Example:**   `set idle 60`

lid_out_addr
>   The lid_out_addr is the name of a dial circuit between two routers. When more
>   than one circuit is configured between two routers (parallel circuits), then there
>   needs to be a way to unambiguously know which dial circuit connects between

them. For this purpose, a lid_out_addr is sent from the router at one end (the caller). At the receiving end the other router configures the same string as the inbound destination name. The lid_out_addr must be an address name that has previously been added using "ADD ISDN-ADDRESS" from the config> prompt.

priority
> The priority field allows an outbound dial-on-demand circuit to preempt another when no channels are available. If a call request is made and all the channels are in use, then the priority of the requesting dial-on-demand circuit is checked against all the active dial-on-demand circuits. If there is an outbound dial-on-demand circuit with lower priority, then that circuit is disconnected and a call is made for the higher priority dial-on-demand circuit. Only the priority on the outbound end of a connection is considered. An inbound dial-on-demand call will not be taken down in favor of a higher priority outbound call. An inbound dial-on-demand call cannot cause a lower priority call to be taken down.

selftest-delay *# of milliseconds*
> Use this parameter to delay the time between when the call is established and the time when the initial packet is sent. Setting a *selftest-delay* can prevent initial packets from being dropped. The range is 0 to 65535, and the default is 150.

> For V.25 bis dial circuits, adjust this setting if your modems take extra time to synchronize.

> For ISDN dial circuits, you may need to adjust this setting for dial-on-demand links because some ISDN switches start to deliver data before signalling the complete establishment of the circuit at the destination end.

Exit

Use the **exit** command to return to the Config> prompt.

Syntax: exit

Example: `exit`

Chapter 40. The WAN Reroute Feature

This chapter describes the WAN reroute feature. It includes the following sections:

- "WAN Reroute Overview"
- "Configuring WAN Reroute" on page 40-2

WAN Reroute Overview

WAN reroute lets you set up an alternate route so that if a primary link fails, the router automatically initiates a new connection to the destination through the alternate route.

The WAN reroute process involves:

1. Detecting the primary link failure
2. Switching to the alternate link
3. Detecting the primary link recovery
4. Switching back to the primary link

You can use any type of link as the alternate link and any type of link as the primary link. The alternate link does not need to be connected to the same end point as the primary link. Figure 40-1 shows how WAN reroute provides an alternate connection between LAN 1 and LAN 2.

LAN 1

Router A

Primary Link

Router B - - - - - - - - - Router C
Alternate
Link

LAN 2   LAN 3
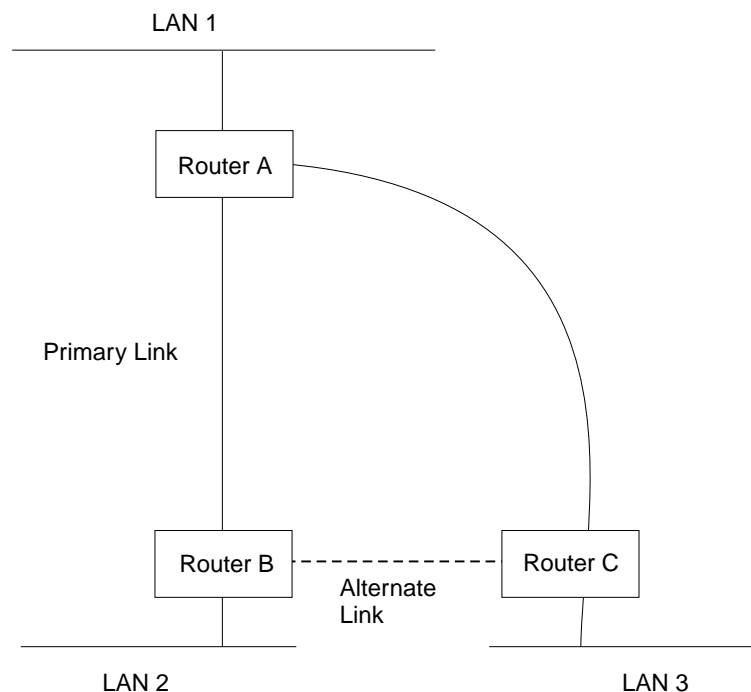
Figure 40-1. WAN Reroute.   Normally, there is a connection between Routers A and B and Routers A and C. If the primary link between routers A and B fails, WAN reroute establishes an alternate link between routers B and C. Routers A and B can then communicate through router C.

Configuring WAN Reroute

Following are the steps required to configure WAN reroute. The next section shows an example of how to perform these tasks.

To configure WAN Reroute, you need to:

1. Configure the primary link.
2. Configure the alternate link.
3. Assign the alternate link to the primary link. You can also specify a stabilization period for the primary link.

    You can specify a time-of-day revert-back to the primary link which will happen after the stabilization period is over (if configured). This allows the secondary to stay up until such time that the user desires and revert back to the primary during off-peak hours.

    Note: The primary and alternate links can be different data-link types (for example, PPP, Frame Relay, or X.25).

Sample WAN Reroute Configuration

Figure 40-2 on page 40-3 shows WAN reroute using a Frame Relay dial circuit over ISDN as the alternate link. If the Frame Relay DLCI between router A and router C fails, WAN reroute uses the dial circuit to establish an alternate connection through router D.
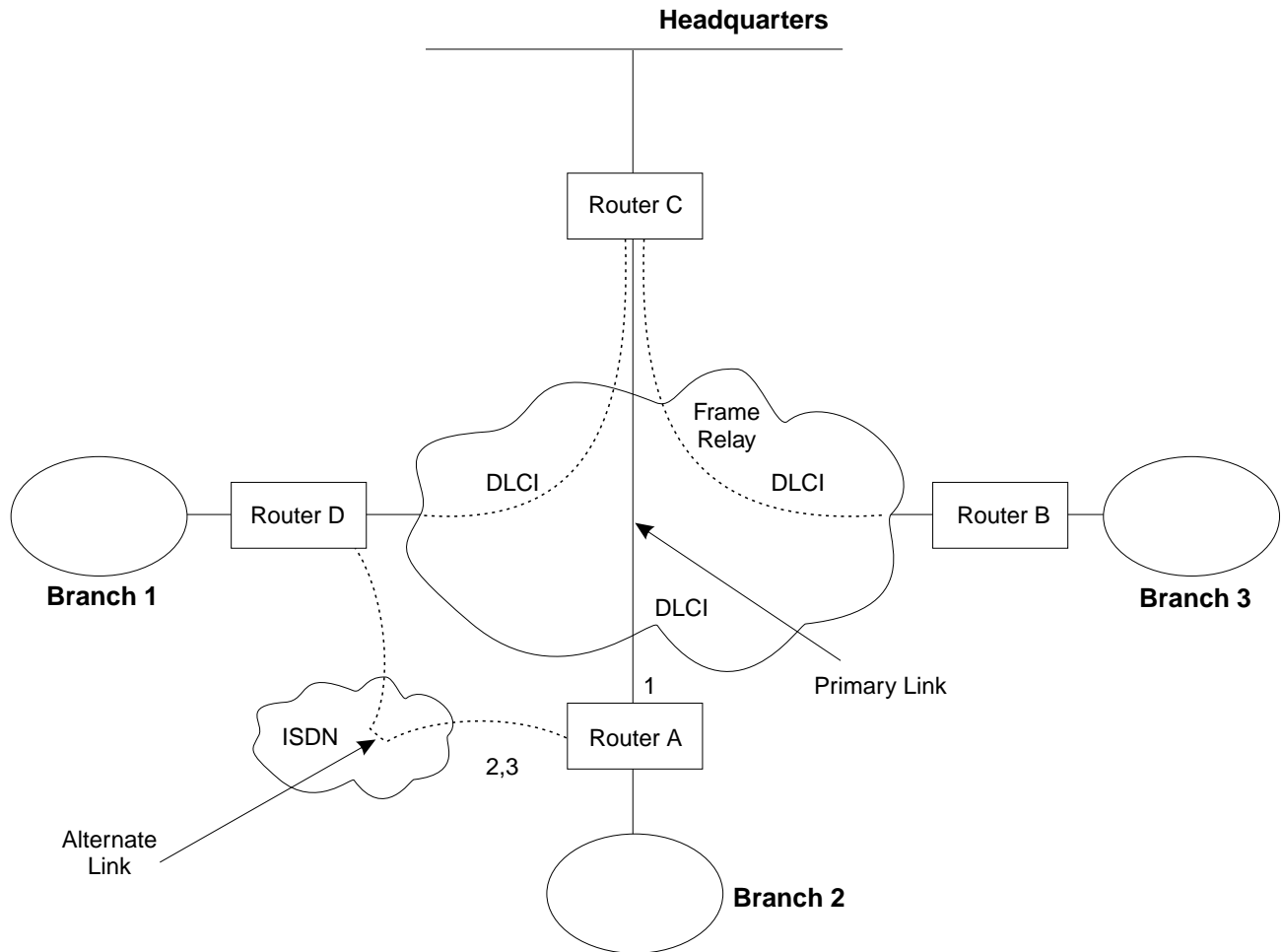
Headquarters

Figure 40-2. Sample WAN Reroute Configuration. Branch offices use frame relay to connect to headquarters. If one of the primary links from a branch to headquarters fails, WAN reroute establishes an alternate route to headquarters through another branch.

The following sections describe how to set up WAN reroute on Router A in Figure 40-2. You will need to:

- Configure the primary frame relay interface (1) to have a Required PVC or Required PVC Group or enable the No-PVC feature on the frame relay interface.

- Configure the ISDN interface (2) and its frame relay dial circuit (3).

- Assign the dial circuit to be the alternate link for the primary frame relay interface.

    Optionally, you can assign:
    – Stabilization period for the primary link,
    – Time-of-day revert-back window for the primary link.

These tasks are described in detail below.

Configuring the Frame Relay Interface

To configure the frame relay interface for WAN reroute, on Router A, add a PVC between Routers A and C on the primary Frame Relay interface.

To cause the primary FR interface to declare itself down when the connection to other router(s) is lost, you have three options:

1. Enable the No-PVC feature. When this feature is enabled, the FR interface goes down when there are no active PVCs.

2. Configure a PVC as required but don't include the PVC in a required PVC group. In this case, the FR interface goes down when the PVC becomes inactive.

3. Configure a set of PVCs as required and as part of a required PVC group. In this case, the FR interface goes down when all of the PVCs of a required PVC group become inactive.

Follow these steps to configure the primary frame relay interface:

1. If you have not yet done so, set the data link on the interface to frame relay.

   ```
   Config>set data-link frame relay
   Interface Number [0]? 2
   ```

2. Enter the Frame Relay configuration process.

   ```
   Config>network
   What is the network number [0]?2
   Frame Relay user configuration
   FR Config>
   ```

   Note: Complete only *one* of the two remaining steps for configuring the primary frame relay interface.

3. Add a PVC using the **add permanent-virtual-circuit** command.

   To configure the PVC as Required:

   > Enter **y** to the question "Is circuit required for interface operation ?."

   To configure the PVC as a member of a required PVC group:

   a. Enter **y** to the question "Does circuit belong to a Required PVC group ?."

   b. Enter a group name in response to the question "What is the group name ?."

   If you have already added PVCs, use the **change permanent-virtual-circuit** command to configure the PVC as Required and to assign it to a Required PVC Group, as appropriate. Refer to Chapter 26, "Using and Configuring Frame Relay Interfaces" on page 26-1 for more information.

   ```
   FR Config>add permanent-virtual-circuit
   Circuit number [16]?
   Committed Information Rate (CIR) in bps [64000]?
   Committed Burst Size (Bc) in bits [64000]?
   Excess Burst Size (Be) in bits [0]?
   Assign circuit name []?
   Is circuit required for interface operation [N]?y
   Does the circuit belong to a required PVC group [N]? y
   What is the group name []?group1
   ```

4. If desired, enable the No-PVC feature.

   Note: Complete this step *only* if you bypassed the previous step.

   ```
   FR Config>enable no-pvc
   ```

There are additional parameters that you can set for frame relay. For more information, see Chapter 26, "Using and Configuring Frame Relay Interfaces" on page 26-1.

Configuring the ISDN Interface and Dial Circuit

Configure the ISDN interface and dial circuit between Router A and Router D. See Chapter 37, "Using and Configuring the ISDN Interface" on page 37-1 for information on how to configure ISDN interfaces and dial circuits.

Unlike WAN restoral, you must configure routing protocols on the dial circuit that will be used as the alternate link. If those routing protocols cannot be prevented from sending maintenance packets, the alternate link will establish a connection even if rerouting is not necessary. In this case if you want to use the alternate link only for rerouting, disable the dial circuit. To disable the dial circuit, enter the **disable interface** command at the `Config>` prompt.

Otherwise, configure the dial circuit to dial on demand by setting the idle timer to non-zero. Use the **set idle** command at the `Circuit Config>` prompt.

If you have multiple dial circuits assigned to the ISDN interface, you can set a priority for the dial circuits. If all the B channels have active dial circuits on the physical interface and a circuit with a higher priority receives a packet, the lowest priority connection is terminated and the high priority circuit establishes a connection.

You can set the priority to between 0 and 15, where 15 is the highest priority circuit and 0 is the lowest priority circuit. The default priority for new dial circuits is 8. Enter **set priority** at the `Circuit Config>` prompt to change the priority.

Assigning and Configuring the Alternate Link

Enter the WAN reroute configuration process to assign the dial circuit as the alternate link for the frame relay interface and, if desired, to specify the stabilization periods and/or the time-of-day revert-back window.

There are two types of stabilization periods:

- *First stabilization period* is the amount of time the router waits for the primary interface to become active when the router first attempts to bring it up. If, after the first stabilization period, the primary has not come up, WAN reroute brings up the alternate link.

- *Stabilization period* is the amount of time the router waits to be sure the primary link is reliable before it switches from the alternate link back to the primary link.

The time-of-day revert-back window is the delay until a specific time of day when the user desires the switch back to the primary after it is up and any configured stability time has passed.

Using a 24-hour clock, the user specifies the start and stop hours of the revert back window. The secondary stays up and is not taken down until the start hour is reached. If the time of day when the primary comes up is between the start and stop hours (in the window) then the switch to the primary link is immediate after the stability time is up.

Follow these steps to assign and configure the alternate link:

1. Enter the WAN restoral configuration process.

   ```
   Config>feature wrs
   WAN Restoral user configuration
   ```

2. Assign the dial circuit as the alternate link for the primary frame relay interface.

   ```
   WRS Config>add alternate-circuit
   Alternate interface number  [0]? 4
   Primary interface number  [0]? 1
   ```

3. Enable the alternate circuit.

   ```
   WRS Config>enable alternate-circuit
   Alternate interface number  [0]? 4
   ```

4. Optionally, specify a first stabilization period.

   To set the first stabilization period for a specific primary interface, use the **set first-stabilization-period** command. To set a default first stabilization period for all interfaces that do not have specific periods set, use the **set default first-stabilization-period** command.

   ```
   WRS Config>set first-stabilization-period
   Primary interface number [0]?
   First primary stabilization time (0 - 3600 seconds -1=default) [-1]?

   WRS Config>set default first-stabilization-period
   Default first primary stabilization time (0 - 3600 seconds) [0]?
   ```

5. Optionally, specify a stabilization period. To set a stabilization period for specific interfaces use the **set stabilization-period** command. To set a default stabilization period for all interfaces that do not have specific periods set, use the **set default stabilization-period** command.

   ```
   WRS Config>set stabilization-period
   Primary interface number [0]?
   First primary stabilization time (0 - 3600 seconds -1=default) [-1]?
   WRS Config>set default stabilization-period
   Default first primary stabilization time (0 - 3600 seconds) [0]?
   ```

6. Optionally, specify a time-of-day revert-back window.

   To set the start and stop times for specific interface windows use the set start-time-of-day-revert-back and stop-time-of-day-revert-back commands. The default value of zero means no window is configured. The 24-hour clock starts at 1am and ends at 24 midnight. If the start and stop times are the same (but not zero) then the revert back will happen at exactly that hour.

   Following are two examples of setting the revert-back window:

   a. A start time of 23 and a stop time of 3 will give a revert-back window from 11pm until 3am.
   b. A start time of 1 and a stop time of 5 will give a revert-back window from 1am to 5am.

   ```
   WRS Config> set start-time-of-day-revert-back
   Primary interface number [0]?
   Time-of-Day revert back window start (1 - 24 hours, 0 = not configured) [0]?
   WRS Config> set stop-time-of-day-revert-back
   Primary interface number [0]?
   Time-of-Day revert back window stop (1 - 24 hours, 0 = not configured) [0]?
   ```

Chapter 41. Overview of LAN Emulation

Note: See the glossary for definitions of the acronyms and terms used in this chapter.

The router implements the *LAN Emulation Over ATM: Version 1.0 Specification*, which is widely accepted as the industry standard for multivendor multiprotocol interoperability. This chapter introduces basic LAN emulation (LE) concepts in the context of the MSS implementation. It begins by examining the motivation for installing emulated LANs (ELANs).

LAN Emulation Benefits

LAN emulation protocols allow ATM networks to provide the appearance of Ethernet and token-ring LANs. Although LAN emulation does not exploit all of the benefits of ATM, it is useful in migrating to ATM technology and lowering network management costs. It enables you to utilize high-speed ATM links and still protect your software and hardware investments.

Software investments are protected because application interfaces are unchanged (LAN emulation is implemented within the data link control layer, which is below the device driver interface of end stations). Hardware investments are protected with forwarding engines that bridge LAN and ATM networks so that existing adapters and wiring can continue to be used.

LAN emulation allows incremental installation of ATM adapters in stations with high-bandwidth requirements, for example, servers and engineering or multimedia workstations. Physical and logical views of a simple LAN emulation example are illustrated in Figure 41-1 on page 41-2.

Simple LAN Emulation Network


```
                    Physical Network                              Logical Network
```
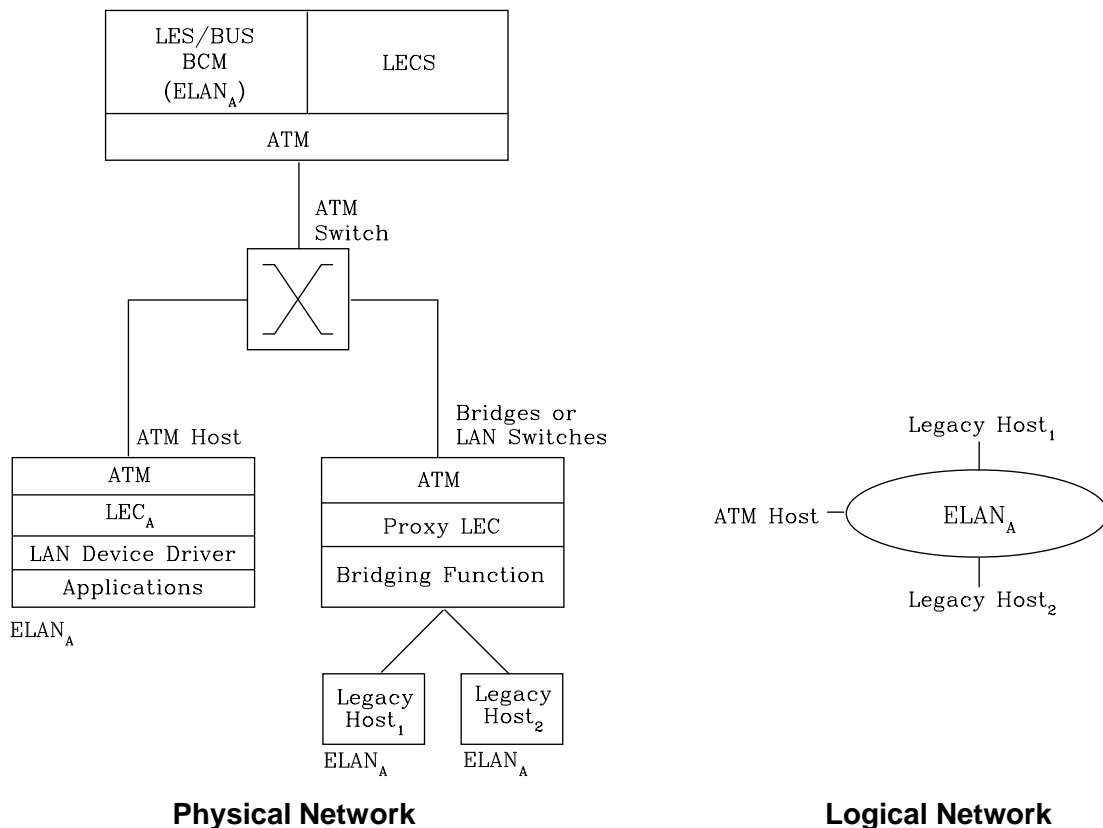
Figure 41-1. Physical and Logical Views of a Simple LAN Emulation Network

The network management benefits of emulated LANs (ELANs) come from increased flexibility in handling moves, adds, and changes. Membership in an ELAN is not based on physical location; instead, logically-related stations are grouped to form an ELAN (stations can also be members of multiple ELANs).

As long as ELAN memberships are retained, no reconfiguration is needed when stations move to new physical locations. Similarly, no wiring modifications are needed to move stations from one ELAN to another.

LAN Emulation Components

The following components implement an ELAN:

LAN emulation (LE) clients
LAN emulation components that represent users of the Emulated LAN.

LE configuration server (LECS)
A LAN emulation service component that centralizes and disseminates configuration data.

LE server (LES)
A LAN emulation service component that resolves LAN destinations to ATM addresses.

Broadcast and Unknown Server (BUS)
> A LAN emulation service component responsible for the delivery of multicast and unknown unicast frames.

The LES, BUS, and LECS are collectively referred to as the LE service components. Each ELAN has a dedicated LES and BUS. LE clients reside in end systems, either in ATM-attached hosts or in bridges or LAN switches. The bridges or LAN switches represent hosts that are connected to Ethernet or token-ring LANs. LE clients provide a MAC-level service to higher level software. Either Ethernet IEEE 802.3 or IEEE 802.5 token-ring LANs can be emulated, but all stations on an ELAN must be of the same type.

The function that bridges between token-ring or Ethernet LAN segments and ELANs is called a Proxy LEC. To emulate a LAN, LE clients request services from the LECS, LES, and BUS. The following sections briefly review ATM addressing and pertinent Interim Local Management Interface (ILMI) functions. You need to understand these concepts before you can understand how the LE components function in the network.

Addressing in ATM

ATM uses 20-byte hierachical addressing as shown below:

```
                                     End System
                                   ┌ Identifier ┐  ┌ Selector
               ┌────── Network Prefix──────┐   │   (ESI)    │  │
                                           │ │            │ │

1                                         13 14          19 20
               ┌───────────────────────────┬─────────────┬──┐
               │                           │             │  │
               │                           │             │  │
               │                           │             │  │
               │                           │             │  │
               └───────────────────────────┴─────────────┴──┘
```
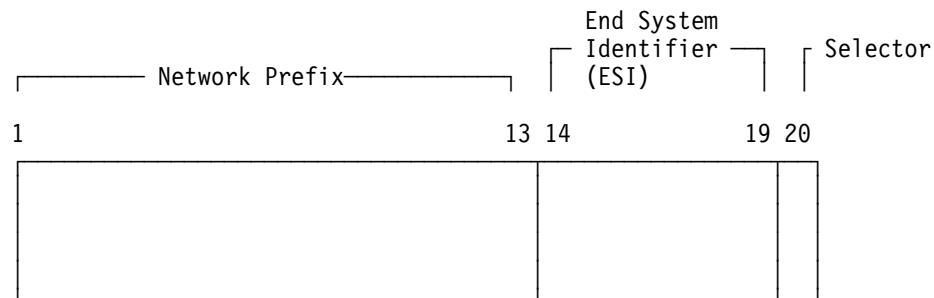
The first 13 octets of an ATM address are the Network Prefix. Each switch in your ATM network must have a unique Network Prefix. ATM switches use the Network Prefix to route VCC setup requests the destination ATM switch. End systems, like this router, retrieve their Network Prefix from their ATM switch when they activate.

Octets 14-19 of an ATM address are the End System Identifier (ESI) Each end system attached to the same switch must use a disjoint set of ESIs. When a end system activates, it attemps to register its ESIs with its ATM switch using the Interim Local Management Interface (ILMI).

ILMI defines a set of SNMP-based procedures used to manage the interface between an end system and an ATM switch. End systems use ILMI to:

- Obtain the network prefix from the switch
- Register their ESIs with the switch
- Dynamically determine the UNI version of the ATM switch
- LECs may get a list of LECS addresses from the switch

The switch forces all of its registered ESIs be unique.

Octet 20 of an ATM address is the selector

End stations obtain their Network Prefix from the switch and form their own addresses by appending an ESI and selector. These addresses must then be registered with the switch, which rejects the registration if the ATM address is not unique

ESI

Each ATM interface on the router has a universally administered, or burned-in, MAC address. The user may use the MAC address as an ESI for some or all of the router's ATM addresses. Alternatively, the user may define up to 64 locally administered ESIs on each interface. If every end system uses its universally administered MAC address as its ESI, then ATM addresses are guaranteed to be unique. This eases the configuration burden. However, using locally administered ESIs can ease problem determination. The user may use any combination of universal or locally administered ESIs.

One way to obtain a unique ATM address is to use a burned-in IEEE MAC address as the ESI and to locally choose a unique selector. By default, the router uses the MAC address of the ATM interface as the ESI in its ATM address(es). Additional ESIs can be configured on each ATM interface.

Each ESI may have up to 255 associated selectors (0x00 through 0xff). The range of selectors is partitioned into two subranges, a configured selector range and an automatically assigned selector range. The ATM interface parameter max-configured-selector gives the upper bound on the configured selector range.

The ATM components on the router have various ways of choosing a selector. Some components force the user to explicitly configure a selector from the configured selector range. LES/BUSs are an example of such a component. Other components, such as Classic IP clients, allow the selector to be automatically assigned at run-time. The user does not have to choose the selector, the router does this when it activates. This selector is not guaranteed to be consistent across router restarts. Automatic selector assignment is only useful for those ATM components whose ATM address does not have to be known apriori by other network devices.

You must configure ATM before you configure emulated LANs, bridging or routing.

ATM Addresses of LAN Emulation Components

In general, ATM addresses must be unique among LAN emulation components. The only exception is that a LES and BUS serving the same ELAN can share an ATM address, as is the case on the router.

LAN emulation components are configured for a particular ATM interface. You can decide to use the burned-in MAC address as the ESI portion of the ATM address of the component or you can select one of the locally-administered ESIs that have been defined for the ATM interface. Multiple LE components can share the same ESI if they have unique selectors. By default, the configuration interface assigns each LE component a unique selector value for the configured ESI; however, you can override this assignment and explicitly configure a particular selector value.

An ATM interface parameter determines the number of selectors per ESI reserved for explicit assignment. The remainder are available for dynamic assignment by the

ATM interface at run-time. LE components use only the selectors reserved for explicit assignment; by default, 200 of the 256 possible selectors per ESI are reserved for explicit assignment. Run-time selector assignment is beneficial when you do not need to control the assigned selector, for example, when you are configuring clients in Classical IP that are not paired with an ARP server.

While ATM addresses must be unique among LE components, LE components can use the same ATM addresses as non-LE components, such as Classical IP servers.

Overview of Related ILMI Functions

ILMI defines a set of SNMP-based procedures used to manage the user-network interface (UNI) between an ATM end system and an ATM switch. The following three ILMI functions are particularly relevant to LAN emulation:

1. ATM address registration, which is described in "Addressing in ATM" on page 41-3

2. Dynamic determination of the signaling version being run at the switch

3. Acquisition of the LECS ATM addresses

As mentioned in "Addressing in ATM" on page 41-3, ATM address registration is a joint effort between ATM end systems and switches. ATM addresses must be registered with the switch before calls can be placed or received.

By default, the ATM interfaces of an router use ILMI procedures to query the switch MIB in an attempt to determine the signaling version (UNI 3.0 or 3.1) being run at the switch. If the query succeeds, the ATM interface runs the same UNI version as the switch; if the query fails, the ATM interface runs UNI 3.0. Alternatively, you can override the default and explicitly configure the UNI version that will run on the ATM interface.

Manual Configuration of the Signaling Version

You need to configure the signaling version manually if the ATM switch runs UNI 3.1 and has no UNI Version MIB variable. In this case, the ATM interface cannot dynamically determine the UNI version. Because the ATM interface in the router uses UNI 3.0 by default, you should manually configure the ATM interface to use UNI 3.1.

Locating the LECS Using ILMI

ILMI is the method of choice for locating the LECS. The ILMI MIB at the ATM switch includes a list of LECS ATM addresses that can be retrieved by LE clients. This method is useful because the LECS ATM addresses need only be configured at ATM switches, not at LE clients, and there are fewer switches than LE clients. Clients attempt to connect to the first LECS on this list. If the connection fails, they try the next LECS address in succession until a connection is established.

Overview of the LECS Function

LE clients are not required to use the LECS, although it is recommended. If the LECS is not used, each LE client must be configured with the ATM address of the LES that serves its ELAN. The LECS reduces the network management burden by serving as a centralized repository for configuration data, minimizing configuration of the LE clients.

Note: At most, one LECS is configurable on each router.

Clients connect to the LECS using well-defined procedures. The following steps are attempted by a client, in order, until a virtual channel connection (VCC) to the LECS is established:

1. Connect to the LECS using any configured LECS address information (configuration of an LECS ATM address at LE clients is optional and is **not** recommended).

2. Obtain a list of LECS addresses using ILMI and attempt to connect to each LECS on the list, in order, until a VCC is established.

3. Establish a VCC to the well-known LECS ATM address as defined by the ATM Forum.

As previously stated, ILMI is the preferred method for LE clients to locate the LECS. The well-known LECS address is needed because some switches do not support the ILMI method. Configuring the LECS address at the LE clients should be done *only* when the switch does not support the ILMI method and the LE service does not support the well-known LECS address.

The router and the IBM ATM switch support all three methods: the pre-configured LECS address, ILMI connection, and the well-known LECS ATM address.

The LECS must provide initial configuration data to LE clients. The most crucial piece of data is the ATM address of the LES. To provide this information to an LE client, the LECS must be able to identify the LE client and to determine the proper LES for that LE client. The LECS identifies a LE client using information in the LE_CONFIGURATION_REQUEST frame sent by the LE client. The configuration request can also contain information to identify the ELAN that the LE client is seeking to join. The following information can be included in the configuration request:

1. Primary ATM address of the LE client

   This field is required and uniquely identifies the LE client.

2. LAN destination associated with the LE client

   This field can contain a MAC address or a route descriptor that uniquely identifies the LE client or it can be unspecified.

3. ELAN Name

   This field can contain a name identifying the requested ELAN or the requesting LE client. In the router implementation, ELAN names are standard ASCII strings. The ELAN name can be unspecified in the request.

4. ELAN Type

This field can specify that the LE client belongs to an Ethernet or token-ring ELAN, or it can be unspecified. If the LE client specifies the type of ELAN, the LECS cannot assign the client to an ELAN of a different type.

5. Maximum frame size supported by the LE client

This field can specify the upper bound on the size of a data frame that can be processed by the LE client, or it can be unspecified. The LECS cannot assign a client to an ELAN with a maximum frame size *larger* than that specified by the client. If the ELAN allows frames too large for the client to handle, the client cannot function on that ELAN.

Given this information, the LECS assigns the LE client to a LES. This is accomplished through the use of policies and policy values. A policy is a criterion that the LECS uses to make LE client-to-LES assignment decisions. A policy value is a (value, LES) pair that indicates that the specified value should be assigned to the specified LES. For example, a policy could be the MAC address of the LE client, and a policy value could be (MAC ADDR_A, LES_1). A LE client with MAC ADDR_A will be assigned to LES_1 if the LE client has not already been assigned to another LES because of a higher-priority policy. One set of policies and policy values applies to all the ELANs.

In accordance with the LE service MIB Specification of the ATM forum, these are the six policies defined:

1. ATM address
2. MAC address
3. Route descriptor
4. ELAN type
5. Max frame size
6. ELAN name

Policies also have priorities. The LECS examines policies in prioritized order. Policies with smaller values in the priority field are considered before policies with larger values in the priority field. Policies with equal values in the priority field are considered at the same time and *ANDed* together.

The LECS assigns a LE client to a LES when all of the policies at the current priority level are satisfied and in agreement. The policies are satisfied when there is a policy value that matches the corresponding field in the configuration request for each policy at the current level. The policies are in agreement when the set of matches include a LES that is common to all the policies. If these conditions are not met, the LECS considers the policies at the next priority level. If the LECS is unable to find a LES at any priority level, an unsuccessful configuration response is returned to the LE client.

To understand the meaning of agreement of the policies, consider this example of policies not in agreement. Suppose that the policies at priority one are a MAC address and an ELAN name. One of the policy values is (X'400000121225', LES_A) and one is (ELAN 1, LES_B). If the LE client provides a LAN destination of X'400000121225', the MAC address policy is satisfied. If the LE client provides an ELAN name of *ELAN 1*, then the ELAN name policy is also satisfied. In this case the policies at priority one are *NOT* in agreement because they refer to different LESs. In this example, the LECS would examine the policies at the next priority level.

After determining the correct LES for an LE client, the LECS returns a configuration response to the LE client that includes the following information: LES ATM address, ELAN type, max frame size, and ELAN name. The configuration response can also include type/length/value (TLV) parameters. TLVs provide a method to download optional or user-defined parameters to the LE client.

Sample Situations for Use of the LECS Assignment Policies

The following section offers examples of various LECS assignment policies.

ATM Address Policy

The MSS Server LECS permits two types of ATM address policy values. The first type is a variable length ATM address prefix. For example, the policy value (399999999999900000102, LES_A) implies that all LE clients whose ATM address begins with 399999999999900000102 should be assigned to LES_A.

The second type of ATM address policy value is an ESI and Selector of an ATM address. For example, the policy value (10002345003281, LES_A) implies that the LE client with an ESI of 100023450032 and a selector of 81 should be assigned to LES A.

When given the ATM address of an LE client, the LECS searches first for a matching ESI and selector. If no match is returned, the LECS searches for the ATM address prefix policy value with the longest matching prefix. Thus, for example, the above policy value value (399999999999990000, LES_B).

ATM address ESI and selector policy values can be used to assign clients to LESs in a manner independent of the LE clients physical location (the ESI and selector is defined locally to the client). ATM address prefixes are the only policy values which indicate any geographic information.

LAN Destination Policy

LE clients may be assigned to LESs based upon a MAC address or a route descriptor. Because a LAN destination uniquely identifies a LE client in a manner that is independent of geographic location, this policy is useful in ensuring that the LE client is assigned to the proper ELAN regardless of its physical location, for example, retaining the ELAN memberships of a workstation when it is moved from one switch to another.

ELAN Name Policy

ELAN names are perhaps the most flexible of the assignment criteria. Some of the ways that ELAN name policy values can be used are:

- Use the actual name of the ELAN

  If LES_A serves Elan 1, then create the policy value (Elan 1, LES_A). LE clients specifying Elan 1 in configuration requests will then be assigned to LES_A.

- Use aliases for the ELAN

  For example, all LE clients belonging to members of the Accounting Department could be configured to use the ELAN name *Accounting*, while those belonging to the Engineering Department could use the ELAN name *Engineering*. Depending upon the number of LE clients on the ELANs, these names could be directed to the same ELAN:

Configure policy values:

```
(Accounting,  LES_A)
(Engineering, LES_A)
```

or to different ELANs:

Configure policy values

```
(Accounting,  LES_A)
(Engineering, LES_B)
```

This setup requires configuring the LE clients with the correct ELAN Name.

- Use names for the LE clients

  Each LE client can be given its own name. For example, one could create the policy values (Joe, LES_A) and (Mary, LES_A). Then, the LE clients configured with these names would be directed to the same LES. This method requires configuring the ELAN name at each LE client and at the LECS. However, it allows Joe and Mary to move the client to a new location. Even though moving causes the client to have a new ATM address or MAC address, as long as you configure the new LE client with the same ELAN name, you retain membership in the original ELAN. This technique also offers a moderate amount of security if the names of each LE client are considered to be passwords.

ELAN Type Policy

ELAN type policy values are most useful for providing default ELANs. For example, the following policy values would ensure that every LE client is assigned to one of the LESs:

```
(Token-ring ELAN Type,  LES_A)
(Ethernet ELAN Type,    LES_B)
(Unspecified ELAN Type, LES_C)
```

In general, policies used for providing default ELAN assignments should be given a low priority, so that the more specific policies are considered first.

Max Frame Size Policy

The max frame size policy can also be used to provide default ELAN assignments.

Duplicate Policy Values

Duplicates occur when the same policy value is associated with multiple LESs for a given policy. Duplicate policy values are allowed for the ELAN type and max frame size policies, but are not allowed for other policies. Duplicate values are useful only when combined with a different policy of the same priority.

For example, assume that there are 3 ELANs: an Ethernet ELAN with a max frame size of 4544 bytes, a token-ring ELAN with a max frame size of 4544 bytes, and another token-ring ELAN with a max frame size of 18190 bytes. LE clients could be assigned to the appropriate ELAN by setting the ELAN type and max frame size policies to the same priority level and defining the following policy values:

```
(Ethernet ELAN Type,     LES_1)      (Max Frame Size = 4544,   LES_1)
(Token-Ring ELAN Type,   LES_2)      (Max Frame Size = 4544,   LES_2)
(Token-Ring ELAN Type,   LES_2)      (Max Frame Size = 18190,  LES_2)
```

More Information About TLVs

TLVs are defined on an ELAN basis; therefore, the same set of TLVs is returned to all LE clients that are assigned to a particular ELAN. When a TLV is included in a configuration response, the LE client ***must*** use the value specified in the TLV as an operating parameter (if the LE client recognizes the ELAN type). A few examples of situations where TLVs might be beneficial are as follows:

- When ELANs are spread over large geographic locations, the default timeout values for LE clients may be insufficient. These timeouts can be controlled for all LE clients by specifying their value in a TLV at the LECS.

- By default, ELANs use best effort connections to connect to the BUS. For ELANs where BUS traffic is heavy, better performance can be obtained by using reserved bandwidth connections to the BUS. The characteristics of the Multicast Send VCC between the LE client and the BUS can be controlled with TLVs.

- A TLV can be used to download the ELAN segment number to source route bridges.

In addition to fine-tuning the configuration, TLVs force all clients on the ELAN to operate with consistent parameters. The router supports all ATM Forum-defined TLVs along with arbitrary, user-defined TLVs.

Connecting to the LES

After obtaining the ATM address of the LES, the LE client initiates a Control Direct VCC to the LES. When this VCC has been established, the LE client sends an LE_JOIN_REQUEST to the LES. The LES responds by adding the LE client to the appropriate point-to-multipoint Control Distribute VCC and returning an LE_JOIN_RESPONSE. By default, the LES partitions proxy and non-proxy clients onto separate Control Distribute VCCs as illustrated in Figure 41-2; however, you can configure the LES to use a single Control Distribute VCC for all LE clients in order to reduce the number of point-to-multipoint VCCs that are required. Partitioning the VCCs is generally useful because it reduces the amount of nuisance traffic that is sent to non-proxy clients. No LE_ARP_REQUESTs are sent to non-proxy LE clients, as described in "Address Resolution" on page 41-11.
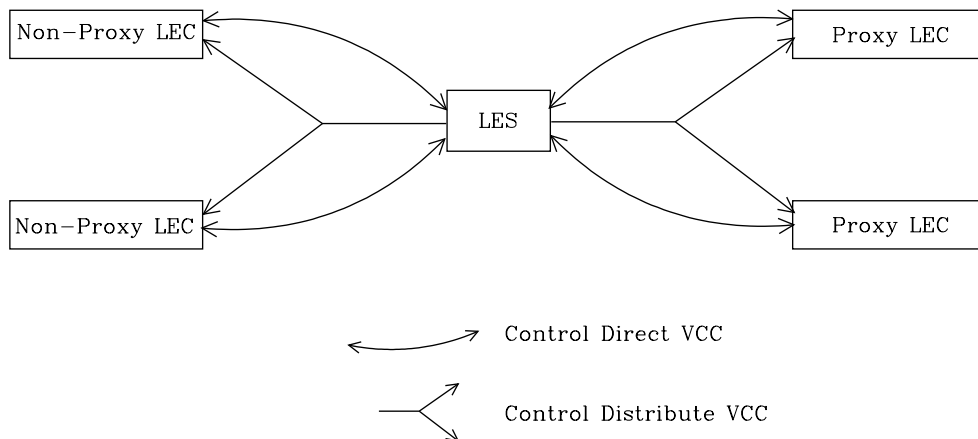
Figure 41-2. Default Connections Between LE Clients and the LES

The following ATM connections are established between the LE client and the LES:

Control Direct VCC (bi-directional point-to-point)
    From LE client to LES

Control Distribute VCC (point-to-multipoint)
    From LES to LE client

Address Registration

LE clients register LAN destinations with the LES to ensure uniqueness and to allow the LES to answer LE_ARP_REQUESTs, which LE clients issue to learn the ATM address associated with a particular LAN destination. Registrations include the LAN destination and the ATM address that the LE client associates with the LAN destination. A LAN destination can be either a MAC address or a route descriptor.

Proxy LE clients do not register the MAC addresses of stations on LAN segments that they are bridging to the ELAN. On the other hand, non-proxy LE clients must register all the LAN destinations that they represent. All route descriptors must be registered, regardless of whether they are associated with a proxy or a non-proxy LE client. Route descriptors are applicable only to proxy LECs that are performing source route bridging. A route descriptor contains the bridge number of the proxy LE client and the segment number of a ring that the LE client is bridging to, that is equivalent to one hop away.

Address Resolution

LAN communications are based upon source and destination MAC addresses. To enable such communication on an ATM network, MAC addresses must be resolved to ATM addresses. An LE client sends an LE_ARP_REQUEST to the LES to learn the ATM address of a particular LAN destination. If the LAN destination is registered, the LES responds with the ATM address associated with the LAN destination. Otherwise, the request is forwarded to all proxy LE clients on the Control Distribute VCC. There is no need to forward the request to non-proxy LECs because all of their LAN destinations are registered; however, if the LES is configured to use a single Control Distribute VCC, both proxy and non-proxy LE clients will receive the request. Control Distribute VCCs provide an efficient way for the LES to distribute control frames to multiple LE clients.

Proxy LE clients respond to LE_ARP_REQUESTs for unregistered MAC addresses that they represent. The LE_ARP_RESPONSE is sent to the LES on the Control Direct VCC, and the LES forwards the response to the LE client that issued the request.

Connecting to the BUS

After connecting to the LES, an LE client issues an LE_ARP_REQUEST for the all 1s broadcast MAC address. The LES responds with the ATM address of the BUS. The LE client then initiates the establishment of a Multicast Send VCC to the BUS. The BUS responds by adding the LE client to the appropriate point-to-multipoint Multicast Forward VCC. By default, the BUS partitions proxy and non-proxy clients onto separate Multicast Forward VCCs; however, as was the case with the Control Distribute VCC, you can configure the BUS to use a single Multicast Forward VCC

for all LE clients. Figure 41-3 on page 41-12 shows partitioned Multicast Forward VCCs.
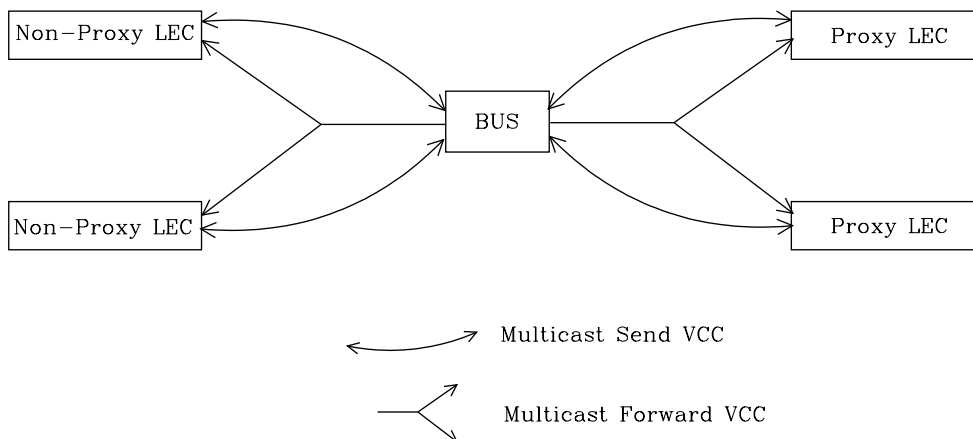
Figure 41-3. Default Connection Between LE Clients (LECs) and BUS

This list is provided to help you clarify the ATM connections that are established between the LE client and the BUS:

Multicast Send VCC (bi-directional point-to-point)
    From LE client to BUS

Multicast Forward VCC (point-to-multipoint)
    From BUS to LE client

BUS Functions

The BUS has two basic functions:

1. Distribute multicast frames to all the LE clients in the ELAN
2. Forward unicast frames to the appropriate destinations

A LE client sends unicast frames to the BUS if it does not have a direct connection to the LE client that represents the destination. To avoid creating a bottleneck at the BUS, the rate at which an LE client can send unicast frames to the BUS is limited.

In the router implementation, the BUS has two modes of operation: partitioning the unicast frame domain and not partitioning the unicast frame domain. If you partition the unicast frame domain, the BUS uses two Multicast Forward VCCs. Otherwise, the BUS uses a single Multicast Forward VCC.

If a single Multicast Forward VCC is used, the BUS operation is very simple; all received frames are simply forwarded to all LE clients. If two Multicast Forward VCCs are used, the BUS will not broadcast unicast frames to all LE clients; instead, unicast frames destined for non-proxy LE clients will be transmitted directly to the destination LE client on a Multicast Send VCC, and all other unicast frames will be transmitted only to proxy LE clients, using the Proxy Multicast Forward VCC. When two multicast VCCs are used, the router is considered to be in intelligent BUS (IBUS) mode.

IBUS mode reduces nuisance unicast frames, which are unicast frames not destined for the client; proxy clients do not receive unicast frames destined for

non-proxy clients, and non-proxy clients never receive nuisance unicast frames. Network bandwidth devoted to nuisance frames is also reduced. On the other hand, BUS processing requirements are increased and multicast frames must be transmitted twice (once on each Multicast Forward VCC). In general, IBUS operation is recommended; however, this option should be disabled in configurations that have source route bridges that join the ELAN as non-proxies.

Establishing Data Direct VCCs

Data Direct VCCs connect two LE clients, and are used to exchange unicast frames without involving the BUS. The LE client uses the address resolution procedures to determine the ATM address associated with the desired LAN destination. If the LE client already has a Data Direct VCC to the ATM address (perhaps for another LAN destination represented by the target LE client), unicast data frames are subsequently transmitted on the existing VCC; otherwise, the LE client invokes the signaling protocol to establish a new VCC.

The LE client maintains an LE_ARP cache containing LAN destination-to-ATM address mappings. Entries in this cache are aged and must be periodically refreshed. The entries are refreshed when a data frame is received from the LAN destination. The LE client also attempts to refresh entries in the absence of data traffic.

Utilization of Data Direct VCCs is also monitored and the VCCs are released if there is no traffic for the VCC time-out period, which is configurable. Additionally, Data Direct VCCs are released in a least-recently used manner when establishment of a new Data Direct VCC fails due to insufficient resource availability.

Overview of router Extensions for LAN Emulation

IBM has made value-add extensions to ATM Forum LAN Emulation available on the router. These extensions offer improved performance, reliability, security, and manageability:

Broadcast Manager (BCM)
> This function can improve overall network performance by reducing ELAN broadcasts.

Redundancy
> The redundancy mechanism improves reliability by allowing backup servers to take over if failures occur at primary servers.

Security
> Security is improved by letting the LECS control ELAN memberships.

BUS Monitor
> This function enhances manageability by identifying the top users of the BUS.

The following four sections describe each of these extensions.

Broadcast Manager (BCM)

BCM is an extension to LAN emulation that consists of IBM enhancement of the LAN emulation BUS. Without BCM, the following events occur:

- A multicast frame sent to the BUS is forwarded to all LE clients on the ELAN.

- LE clients that include the proxy function to provide bridging support forward the broadcast frame onto other LAN segments.

- All end stations receive and process every broadcast frame.

BCM can be enabled on individual ELANs for any of these protocols:

    IP
    IPX
    NetBIOS

When BCM is enabled, a minimal amount of layer 2 and layer 3 information is decoded for specific types of broadcast frames sent to the BUS. Whenever possible, BCM transforms broadcast frames into unicast frames, and sends them only to interested LE clients and end stations. BCM reduces both network traffic and associated end-station overhead by filtering nuisance broadcast frames. These functions can improve overall system performance and enable practical deployment of larger ELANs.

BCM Support for IP

When enabled for IP, BCM scans all IP ARP requests and replies to learn the location of IP addresses in the IP subnet that contains this ELAN. The objective is for BCM to take each broadcast ARP request frame and forward it as a unicast frame directly to the LE client representing the target IP station. Both network traffic and end-station processing time are reduced when the request is forwarded directly to the appropriate LE client on the Multicast Send VCC instead of being broadcast to all LE clients on the Multicast Forward VCCs. When the destination station is located behind a bridge function, the LAN that the destination station belongs to also benefits from the reduced broadcast traffic.

BCM Support for IPX

For IPX, BCM limits the scope of advertisement broadcasts. IPX routers and servers periodically broadcast their known network and service information. Generally, these broadcasts, called Routing Information Protocol (RIP) and Service Advertising Protocol (SAP) general response packets, need to be received only by other IPX routers and servers.

When it is enabled for IPX, BCM dynamically identifies the set of IPX routers and servers based on advertisement transmissions, and only forwards RIP and SAP advertisements to other IPX routers and servers.

With BCM enabled, any quiet device (that is, a device that does not transmit IPX advertisements) that needs to receive IPX advertisements has to be configured as a static target. An example of such a device is a station running software that discovers the IPX network topology by monitoring IPX advertisements.

BCM Support for NetBIOS

NetBIOS is considered to be a broadcast-abusive protocol and therefore an excellent candidate for BCM. NetBIOS communication is based on names. Transmitting stations can learn the MAC address associated with a particular destination name by broadcasting a query or by having the frame multicasted to the NetBIOS functional address. In the latter case, every NetBIOS device in the network must receive the frame and determine whether the destination name on the frame applies to itself. To make things even worse, NetBIOS devices tend to repeat transmission of certain types of frames as much as 10 times. Historically, this was to ensure that all devices receive the frame in cases where the network is heavily congested.

The BCM strategy is to associate unique NetBIOS names with MAC addresses and LE clients by learning names from NetBIOS frames sent to the BUS. After a unique NetBIOS name is learned, subsequent NetBIOS broadcast frames destined for that name are forwarded to a single LE client as a unicast frame. BCM also filters certain NetBIOS frames that are broadcast repeatedly.

BCM Support for Source Route Bridging

Source Route Management (SRM) is an additional BCM feature that can be configured for 802.5 ELANs. When enabled, this feature will further process frames managed by BCM and, whenever possible, transform All Routes Explorer (ARE) or Spanning Tree Explorer (STE) frames into Specifically Routed Frames (SRF). Once a frame is transformed into a SRF, the frame no longer needs to be transmitted onto each ring in the bridged network.

The token-ring topology behind each LE client is learned by recording the routing information field (RIF) of frames received by the BUS. Because SRM dynamically learns token-ring topology information, an aging mechanism is used to remove information that has not been refreshed recently.

To decide whether to enable BCM or SRM (or both), you should compare the net system-wide benefit with the inevitable reduction in the rate at which packets are forwarded when BCM or SRM is enabled.

Note: Broadcast Manager and Source Route Management are unavailable and cannot be enabled if **bus-mode** is set to *adapter*.

LAN Emulation Reliability

A perceived lack of robustness has been one of the most widely proclaimed criticisms of LAN emulation. While the ATM Forum is addressing this issue with specifications for distributing the LE service, the router offers an answer in the interim. Figure 41-4 on page 41-16 provides a framework for describing the MSS solution.
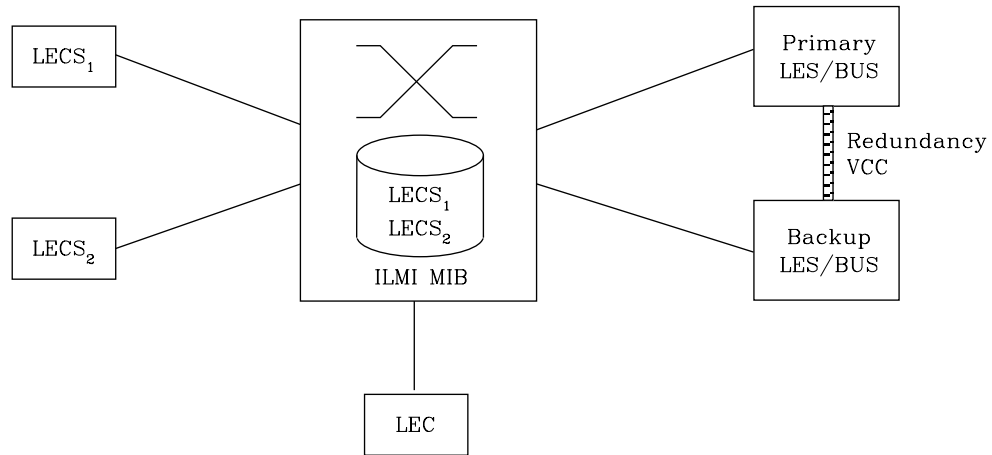
Figure 41-4. LAN Emulation Redundancy

Each LES/BUS may be independently configured for redundancy (the default is no redundancy). If redundancy is enabled, the LES/BUS is configured to assume the role of a primary or a backup LES/BUS. Unless it has been configured as a redundant LES/BUS, the LES/BUS is primary. The primary LES/BUS is typically the only LES/BUS visible to the LE clients. It is responsible for setting up and maintaining a Redundancy VCC to the backup LES. The presence of this VCC indicates that the primary LES/BUS is operational. The backup LES will not accept Control Direct VCC calls while the Redundancy VCC is established. However, if the Redundancy VCC is **not** present, the backup LES/BUS services ELAN requests in the usual manner.

For the redundancy protocol to be effective, LE clients must detect the failure of the primary LES/BUS and connect to the backup. LE clients detect server failures by means of released VCCs. Connection to the backup LES/BUS is accomplished through the LECS.

Upon receiving an LE_CONFIGURE_REQUEST, the LECS assigns the LE client to the appropriate LES and ELAN. If this LES has no configured backup, then the LECS returns the ATM address of the LES. If the LES is configured with a backup LES, then the LECS may return the primary or backup LES address. The backup LES address is returned if the backup LES exists on the same MSS Server as the LECS and is currently serving the ELAN, if the primary LES exists on the same MSS Server as the LECS and it is not currently serving the ELAN, or if neither LES exists on the same MSS Server as the LECS and the client was assigned to last assigned to the primary LES (within the past 5 minutes). Otherwise, the primary LES address is returned to the LE client.

The LECS retains a short-term memory of all client assignments so that it may alternately direct a LE client to a primary and backup LES. This simple heuristic makes the correct assignment in the nominal case of no failure and is self-correcting. At worst, the heuristic causes the LE client to repeat the configuration phase of joining an ELAN.

LECS robustness can be achieved by establishing duplicate LECSs on multiple platforms and including their ATM addresses in the ILMI database. LE clients will then connect to the backup LECS if the primary is unavailable.

LAN Emulation Security

Traditional LANs offer security in the sense that a physical connection implies that two stations are on the same LAN. Because multiple emulated LANs can exist on a single ATM network, stations that are not on the ELAN can be physically connected to stations that are on the ELAN. This situation presents a security risk in that unauthorized stations can connect to the LES and attempt to use its services.

To control ELAN membership, an MSS LES can be configured to validate LE_JOIN_REQUESTs with the LECS. In this mode the LES forms an LE_CONFIGURE_REQUEST on behalf of the LE client using information from the LE_JOIN_REQUEST. These LE_CONFIGURE_REQUESTs include the source LAN destination, Source ATM address, ELAN type, max frame size, and ELAN name from the LE_JOIN_REQUEST, along with an IBM Security TLV. The security requests are transmitted to the LECS by a multiplexing component called the LECS interface, and the LECS must validate the requests using its ELAN assignment database before LE clients are allowed to join the ELAN.

A LECS interface is associated with an ATM interface, and all LESs configured on the ATM interface use the same LECS interface. The LECS interface conserves VCC resources by multiplexing security requests from multiple LESs onto a single VCC to the LECS. The LECS interface locates the LECS dynamically using the ILMI and well-known LECS address mechanisms. After the VCC to the LECS is established, the LECS interface issues a local query to determine whether the LECS is located on the same router. If the LECS is located on the same router, a local interface is used to confirm requests to join without transmitting requests onto the ATM network.

To maximize the security of an ELAN, the following steps are recommended:

1. At the LECS, use ATM addresses to assign clients to the LES. See "Overview of the LECS Function" on page 41-6 for more information.

2. Activate the LECS Interface on the router.

3. Activate the security option of the LESs.

4. Use *Address Screening* at the ATM switches. This option causes switches to validate that calling stations use their actual ATM addresses in the call setup. Thus, stations cannot impersonate other stations.

These steps ensure that stations are correctly identified and that only authorized stations join the ELAN.

BUS Monitor

The BUS Monitor provides a way to pinpoint end users that can be over-utilizing the BUS. When enabled, the BUS Monitor periodically samples the traffic sent to the BUS on a particular ELAN. At the end of each sample interval, the BUS Monitor identifies the top users of the BUS by their source MAC addresses, LE client ATM addresses, and the number of sampled frames each of them has sent to the BUS. You can configure the following parameters for the BUS monitor:

* The number of MAC addresses (hosts) to record as top users

* The number of seconds in each sample interval

- The sample rate. The sample rate consists of the fraction of all the frames received that the sample consists of, for example, 1 out of every 100 frames, 1 out of every 10 frames, or every frame.

- The number of minutes between sample intervals

Key Configuration Parameters for LAN Emulation

This section briefly describes the required configuration parameters of the router LAN emulation components. The ATM interface for the LAN emulation components must be defined before the components can be created.

1. LEC:

   To create a LE client, the user only needs to specify the ELAN type. If two LE clients are defined on a single ATM interface and are bridged together, then one of the LE clients must use a non-default MAC address. By default, LE clients use the burned-in MAC address of the ATM interface. The default maximum frame size is 1516 bytes for Ethernet LE clients and 4544 bytes for token-ring LE clients.

2. LES/BUS:

   The required parameters for a LES/BUS are the ELAN name, the ELAN type, and the ESI (which is selected from a list that includes the burned-in MAC address and any locally-administered values defined for the ATM interface). Defaults are supplied for other parameters.

   The maximum frame size default is 1516 bytes for Ethernet ELANs and 4544 bytes for token-ring ELANs. LE clients will not be allowed to join the ELAN if their maximum frame size is less than the maximum frame size of the ELAN; LE clients that have a larger maximum frame size will be allowed to join the ELAN, but will use the maximum frame size of the ELAN as a result of join-time negotiation with the LES.

3. LECS:

   At a minimum, the LECS ESI must be selected and a default ELAN assignment policy must be configured. See "Overview of the LECS Function" on page 41-6 for more information.

Chapter 42. Using and Configuring ATM

This chapter describes how to configure the ATM interface. It includes the following sections:

- "ATM and LAN Emulation"
- "Accessing the ATM Interface Configuration Process" on page 42-2
- "ATM Configuration Commands" on page 42-3

ATM and LAN Emulation

LAN emulation provides support for virtual Token-Ring and Ethernet LANs over an ATM network. Refer to "Addressing in ATM" on page 41-3 for a discussion of ATM addressing.

How to Enter Addresses

Addresses are entered in two ways, depending upon whether the address represents (1) an IP address, or (2) an ATM address, MAC address, or route descriptor, as follows:

1. IP address

   IP addresses are entered in dotted decimal format, a four-byte field represented by four decimal numbers (0 to 255) separated by periods (.).

   Example of IP Address:

   ```
   01.255.01.00
   ```

2. ATM or MAC address or route descriptor

   ATM addresses, MAC addresses, and route descriptors are entered as strings of hexadecimal characters with or without optional separator characters between bytes. Valid separator characters are dashes (–), periods (.), or colons (:).

   Examples of ATM address, MAC address or route descriptor:

   ```
   A1FF010203
        or
   A1–FF–01–02–03
        or
   A1.FF.01.02.03
        or
   39.84.0F.00.00.00.00.00.00.00.00.00.03.10.00.5A.00.DE.AD.08
        or
   A1:FF:01:02:03
        or even
   A1–FF.01:0203
   ```

   Each type of address requires a different number of hexadecimal characters:

   ATM   40

   MAC   12

   ESI   12

   Route descriptor 4

   This information applies to addresses entered for ATM; LAN emulation; Classical IP and ARP over ATM; and IPX and ARP over ATM.

Accessing the ATM Interface Configuration Process

Use the following procedure to access the configuration process.

1. At the OPCON prompt, enter **talk 6**. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

   ```
   * talk 6
     Config>
   ```

   After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the CONFIG prompt, enter the **list devices** command to display the network interface numbers for which the router is currently configured.

3. Record the interface numbers. If ATM is not specified as an interface, then create a ATM interface by using the **add device** command at the Config> prompt.

4. Enter the **network** command and the number of the ATM interface you want to configure. For example:

   ```
   Config> network 0
   ATM Config>
   ```

   The ATM configuration prompt (`ATM Config>`), is displayed.

ATM Configuration Commands

This section summarizes the ATM configuration commands. Enter the commands at the ATM config> prompt.

Table 42-1. ATM Configuration Command Summary	
Command	**Function**
? (Help)	Lists all of the ATM configuration commands, or lists the options associated with specific commands.
INTERFACE	Gets you to the ATM Interface Config> prompt from which you can list, change, or configure the ATM Interface. • Add an ESI. • List the current configuration or list ESIs. • Remove an ESI. • Set parameters of the ATM network. • Enable or disable an ESI. • Exit
LE-CLIENT	Gets you to the LE Client Config> prompt from which you can list, change, or configure the LAN Emulation Client Interface as described in Chapter 44, "Using and Configuring LAN Emulation Clients" on page 44-1. • Add a LAN Emulation Client (LEC) for a token-ring or Ethernet emulated LAN. • Configure a network by network #. This command takes you to the LE Config> prompt, from which you can configure a specific LAN Emulation Client (LEC). • List LAN Emulation Clients (LECs). • Remove a LAN Emulation Client (LEC).
Exit	Exit the ATM Configuration process and returns to the Config> prompt.

ATM Interface Commands (ATM INTERFACE> Prompt)

This section summarizes and then explains the commands for configuring a specific ATM interface.

Enter the commands at the `ATM INTERFACE>` prompt.

Table 42-2. ATM INTERFACE Configuration Command Summary

Command	Function
? (Help)	Lists all the ATM Interface configuration commands, or lists the options associated with specific commands.
Add	Adds an ESI.
List	Lists the current configuration or list ESIs.
Remove	Removes an ESI.
Set	Sets parameters of the ATM network.
Disable	Disables an ESI.
Enable	Enables an ESI.
Exit	Returns to the `ATM Config>` prompt.

Add

Use the **add** command to add an ESI to your ATM configuration.

Octets 14-19 of an ATM address are the End System Identifier (ESI) Each end system attached to the same switch must use a disjoint set of ESIs. When a end system activates, it attempts to register its ESIs with its ATM switch using ILMI. The switch forces all of its registered ESIs be unique.

Syntax: <u>a</u>dd   <u>esi</u> *esi-address*

esi *esi-address*
> Address of End System Identifier.

Valid Values: Any 12 hexadecimal digits
Default Value: none

`ATM INTERFACE>` **add esi 014617183763**

List

Use the **list** command to list the configuration of this ATM device or to list the set of configured ESIs.

Syntax: <u>l</u>ist   <u>c</u>onfiguration

                    <u>e</u>si

configuration
> Lists the ATM device configuration. For an explanation of the listed fields, see "Set" on page 42-5.

> **Example: list**

```
                    ATM Configuration

    Interface (net) number =     0
    Maximum VCC data rate Mbps   =      155
    Maximum frame size    = 9234
    Maximum number of callers =  209
    Maximum number of calls =  1024
    Maximum number of parties to a multipoint call =  512
    Maximum number of Selectors that can be configured  =  200
    UNI Version = UNI 3.0
    Packet trace = OFF
```

ESIs

Lists the ESIs in the ATM configuration.

Example: list esi

```
ATM INTERFACE> list esi

      ESI              Enabled
------------------------  -------
000000000009                YES
000000000100                YES
```

Remove

Use the **remove** command to remove an ESI from your ATM configuration. All
ATM components using this ESI should be reconfigured to use a different ESI. An
ATM component which attempts to use a removed ESI will not activate on the next
router restart.

Syntax: remove   esi *esi-address*

esi *esi-address*
    Address of End System Identifier.

Valid Values: Any 12 hexadecimal digits
Default Value: none

Set

Use the **set** command to specify ATM network parameters.

Syntax: set   max-data-rate
                  max-frame
                  max-config-selectors
                  max-calls
                  max-callers
                  max-mp-parties
                  trace
                  uni-version

max-data-rate
    Sets the default and upper bound for VCC traffic parameters of most LANE
    and CIP connections. For example, this is the default PCR for best-effort
    VCCs initiated by LE Clients. Signalled SCRs and PCRs cannot exceed this
    limit. The default value should be satisfactory in most situations. An example
    of a situation where it is beneficial to change this value would be if the
    majority of the stations use 25 Mbps adapters. In this case, it may be
    desirable to limit the data rate on VCCs to 25 Mbps so that the lower speed
    stations are not overwhelmed with frames from the router. The units for this
    parameter are Mbps.

> **Valid Values:** 25 (Mbps)
> 100 (Mbps)
> 155 (Mbps)
> **Default Value:** 155 (Mbps)
>
> **Example:** `set speed`
>
> ```
> ATM INTERFACE> set speed 155
> ```

max-calls

Sets the maximum number of switched virtual circuits (SVCs) that can exist on this ATM device. Every point-to-point and point-to-multipoint SVC uses system resources. This parameter helps limit the system resources reserved for signalling and switched connections. Increasing this parameter will allow more simultaneous SVCs. However, more system memory for managing these connections will be required.

> **Valid Values:** An integer in the range of 64 - 65535
> **Default Value:** 1024
>
> **Example:** `set max-calls`
>
> ```
> ATM INTERFACE> set max-calls 500
> ```

max-callers

Sets the maximum number of entities on the router that use the ATM interface. Each LEC, Classical IP Client, and 1483 bridge interface qualifies as a user of the ATM interface. Increasing this parameter allows more users of the interface and uses more system memory.

> **Valid Values:** An integer in the range of 64 - 1024
> **Default Value:** 209
>
> **Example:** `set max-callers`
>
> ```
> ATM INTERFACE> set max-callers 25
> ```

max-config-selectors

Sets the maximum number of selectors under your specific control.

The selector is used to distinguish different users on the same end system. VCC setup requests are routed in the following heirarchical fashion: ATM switches route to the destination ATM switch using the Network Prefix, the destination ATM switch routes to the destination end system using the ESI, and the end system notifies the destination user based on the selector.

Each ESI may have up to 255 associated selectors (0x00 through 0xff). The range of selectors is partitioned into two subranges, a configured selector range and an automatically assigned selector range. The ATM interface parameter max-configured-selector gives the upper bound on the configured selector range.

The ATM components on the router have various ways of choosing a selector. Some components force the user to explicitly configure a selector from the configured selector range. LES/BUSs are an example of such a component. Other components, such as Classic IP clients, allow the selector to be automatically assigned at run-time. The user does not have to choose the selector, the router does this when it activates. This selector is not guaranteed to be consistent across router restarts. Automatic selector assignment is only useful for those ATM components whose ATM address does not have to be known apriori by other network devices.

The relative sizes of the selector range can be modified to conform to the types and numbers of ATM users on the router.

Valid Values: 0 - 255 (0x00 - 0xFF)
Default Value: 200

Note: The selector is byte 20 of a 20-byte ATM address.

Example: `set max-config-selectors`

```
ATM INTERFACE> set max-config-selectors 225
```

max-frame
> Sets the maximum number of octets permitted in any data frame sent or received on the ATM interface. System memory is allocated based upon this parameter. Increasing the max-frame requires more system memory, but allows processing of larger frames.
>
> All router entities using the ATM interface must use a maximum frame size less than or equal to the max-frame-size of the ATM interface.
>
> **Valid Values:**   An integer in the range of 16 - 65535
> **Default Value:**   9234
>
> **Example: `max-frame`**
>
> ```
> ATM INTERFACE> set max-frame 1000
> ```

max-mp-parties
> Sets the maximum number of leaves on a point-to-multipoint connection intitiated by the router. This parameter affects system memory allocation. Increasing this value is necessary if the router must set up point-to-multipoint connection(s) to a large number of destinations.
>
> **Valid Values:**   An integer in the range of 1 - 5000
> **Default Value:**   512
>
> **Example: `max-mp-parties`**
>
> ```
> ATM INTERFACE> set max-mp-parties 300
> ```
>
> **Valid Values:**   155 (Mbps)
> **Default Value:**   155 (Mbps)
>
> **Example: `set speed`**
>
> ```
> ATM INTERFACE> set speed 155
> ```

trace
> Sets the packet tracing parameters on the interface. Packet tracing can be enabled or disabled on a range of VPI/VCI values. Common VPI/VCI values to trace are:
>
> - 0/5 for signalling packets
> - 0/16 for ILMI packets.
>
> **Valid Values:**   ON or OFF
> **Default Value:**   ON
>
> You are prompted for the VPI/VCI range you want to trace.
>
> **Beginning VPI Valid Values:** 0 - 255
> **Default Value:**   0
>
> **Ending VPI Valid Values:** 0 - 255
> **Default Value:**   255

> **Beginning VCI Valid Values:** 0 - 65535
> **Default Value:**   0
>
> **Ending VCI Valid Values:** 0 - 65535
> **Default Value:**   65535
>
> **Example: `set trace on`**
>
> ```
> ATM INTERFACE> set trace on | off [:]? on
> beginning of VPI range [0]? 0
> end of VPI range [255]? 0
> beginning of VCI range [0]? 5
> end of VCI range [65535]? 5
> ```

uni-version
    Sets the User Network Interface (UNI) version used by the ATM interface with
    communicating with the attached ATM switch. The UNI versions used by the
    ATM interface and by the ATM switch must match. If the UNI version is
    configured as AUTO, the ATM device attempts to learn the UNI version to use
    from the switch.

> **Valid Values:**   [UNI 3.0|UNI 3.1|AUTO-DETECT|None]
> **Default Value:**   UNI 3.0
>
> **Note:** Must match the ATM switch.
>
> **Example: `set uni-version`**
>
> ```
> ATM INTERFACE> set uni-version 3.0
> ```

Enable

Use the **enable** command to enable an ESI in the configuration of your ATM
device. The ATM interface attempts to register only enabled ESIs when it
activates.

Syntax: enable   esi *esi-address*

esi *esi-address*
    Address of End System Identifiers.

> **Valid Values:**   Any 12 hexadecimal digits
> **Default Value:**   none
>
> **Example: `enable esi`**
>
> ```
> ATM INTERFACE> enable esi 00:00:00:00:00:09
> ```

Disable

Use the **disable** command to disable an ESI in the configuration. ATM
components using disabled ESIs will not become active on the next router restart.

Syntax: disable   esi *esi-address*

esi *esi-address*
    Address of End Sytem Identifiers.

> **Valid Values:**   Any 12 hexadecimal digits
> **Default Value:**   none
>
> **Example: `disable esi`**
>
> ```
> ATM INTERFACE> disable esi 00:00:00:00:00:09
> ```

Exit

Use the **exit** command to return to the `ATM Config>` prompt.

Syntax: <u>ex</u>it

Example: exit

```
ATM INTERFACE> exit
ATM Config>
```

Chapter 43. Monitoring ATM

This chapter describes how to monitor the ATM interface. It includes the following sections:

- "Accessing the ATM Console Commands"
- "ATM Console Commands" on page 43-2
- "ATM Interface Console Commands (ATM INTERFACE+ Prompt)" on page 43-3

Accessing the ATM Console Commands

Use the following procedure to access the ATM console commands. This process gives you access to an ATM's *monitoring* process.

1. At the OPCON prompt, enter **talk 5**. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again. To access the ATM interface console process, do the following:

2. Enter **interface** at the + prompt to display a list of configured interfaces.

3. Record the interface numbers.

4. To access the ATM interface console process, do the following:

   Enter **network** followed by the number of the ATM interface.

   ```
       + network 5
   ATM+
   ```

   The ATM monitoring prompt (ATM+), is displayed.

ATM Console Commands

This section summarizes the ATM console commands for monitoring ATM interfaces. Enter the commands at the `ATM+` prompt.

Table 43-1. ATM Console Command Summary	
Command	**Function**
? (Help)	Lists all of the ATM configuration commands, or lists the options associated with specific commands.
INTERFACE	Gets you to the `ATM Interface+` prompt from which you can monitor the ATM Interface, as described in "ATM Interface Console Commands (ATM INTERFACE+ Prompt)" on page 43-3.
Exit	Exits the ATM Monitoring process and returns to the + prompt.

Interface

Gets you to the `ATM Interface+` prompt, described in "ATM Interface Console Commands (ATM INTERFACE+ Prompt)" on page 43-3.

Exit

Returns to the + prompt.

ATM Interface Console Commands (ATM INTERFACE+ Prompt)

This section summarizes and then explains the commands for monitoring a specific ATM interface.

Enter the commands at the `ATM INTERFACE+` prompt.

Table 43-2. ATM INTERFACE Configuration Command Summary

Command	Function
? (Help)	List all the ATM Interface configuration commands, or list the options associated with specific commands.
List	Lists various options
Trace	Starts/Stops packet tracing on a specified VPI/VCI range. Trace can be viewed by ELS.
Wrap	Starts/Stops a loopback test on the VCC.
Exit	Returns to the `ATM+` prompt.

List

Use the **list** command to list various categories of ATM data.

Syntax: <u>l</u>ist   <u>add</u>resses
                <u>al</u>l
                <u>c</u>ircuit
                <u>u</u>sers
                <u>v</u>ccs

addresses
> Lists the ATM addresses in use on the router The ATM address is displayed as the Network Prefix, the End System Identifier, and the Selector.
>
> **Example: `list addresses`**
>
> ```
> ATM INTERFACE+ list addresses
>                     ATM Address
>        Network Prefix                   ESI          SEL
> -------------------------------------- ---------------- --
> 39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.03
> 39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.02
> ```

all
> Lists all of the following:
>
> - Users
> - Addresses
> - VCCs

users

Lists all users of the ATM interface within the router. ATM users include all LECs, CIP-Clients, and CIP servers.

Example: `list users`

```
ATM INTERFACE+ list users

 UserHandle  FrameSap   ATM Address
 ----------  ----------  -------------------------------------------------------------
 2032AC50    2032AC5C   39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.02
 2030A978    2030A984   39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.03
```

circuit

Lists a the statistics for a particular VC by specifying the particular VCI-VPI pair. You can also specify the circuit on the command line; for example: list circuit 33.

Example: `list circuit`

```
ATM INTERFACE+ list circuit
VPI [0]?
VCI [32]?33

     Frames transmitted   =        2  Bytes transmitted   =        216
     Frames received      =        2  Bytes received       =        216
```

vccs

Lists all the VCCs established by the router. The VCCs may be permanent (PVC) or switched (SVC), point-to-point or point-to-multipoint and each is identified by a unique VPI/VCI. The trace command uses the VPI/VCI value for a VCC to perform packet tracing over a particular VCC.

Example: `list vccs`

```
ATM INTERFACE+ list vccs
```

Conn Handle	Conn Type	VPI	VCI	FrameSap	Frames Transmitted	Frames Received	Bytes Transmitted	Bytes Received
13	P-MP	0	38	2032AC5C	0	0	0	13 0
10	P-P	0	35	2032DC74	0	62	0	5532
9	P-P	0	34	2032DC74	62	0	5532	0
8	P-P	0	33	2032DC74	0	0	0	0
7	P-P	0	32	2032DC74	2	2	216	216
1	SAAL	0	5	0	582	581	5052	7232
2	ILMI	0	16	0	127	126	5551	8631

P-P   point to point VCC
P-MP   point to multipoint VCC
ILMI   Interim Local Management Interface VCC
SAAL   signalling VCC

Trace

Use the **trace** command activate packet tracing over a specified range of VPI/VCI values. You can view trace data by using ELS as described in "View" on page 9-17.

Syntax: <u>t</u>race   <u>l</u>ist
                   <u>on</u>
                   <u>of</u>f

list

Displays the current packet tracing options on the ATM interface.

Example: `trace list`

```
ATM Interface+ trace
on | off | list []? list
Packet trace is ON
Range of VPIs to be traced:        0 -        0
Range of VCIs to be traced:       32 -       39
```

on

Starts packet tracing on all active VCCs within the specified VPI/VCI range.

Example: `trace on`

```
ATM Interface+ trace on
beginning of VPI range [0]?
end of VPI range [0]?
beginning of VCI range [32]?
end of VCI range [65535]? 39
```

off

Stops packet tracing on all VCCs.

Example: `trace off`

```
ATM Interface+ trace off
ATM Interface+ trace list
Packet trace is OFF
```

Wrap

Use the **wrap** command to perform a loopback data test on the ATM interface of the adapter. Wrap can be issued on a per VC basis by specifying VPI-VCI pairs. Data is looped back internally.

You can selectively start a wrap, stop a wrap, or display the current wrap settings.

If you stop or display a wrap, the following statistics will be displayed:

- Wrap transmits
- Wrap receives
- Wrap transmit errors
- Wrap receive errors
- Wrap receive timeouts

For display, the current the current wrap statistics are displayed.

For stop, the final wrap statistics are displayed.

Syntax: <u>w</u>rap   <u>d</u>isplay
                    <u>sta</u>rt
                    <u>sto</u>p

display

Displays the current wrap settings.

start
> Starts the wrap procedure and specifies the VPI-VCI length of pattern and the pattern itself.

Example: `wrap start`

```
ATM Interface+ wrap start
VPI [0]?
VCI [32]?
wrap pattern length [32]?
Enter 32-byte wrap pattern: [ABCDEFGHIJKLMNOPQRSTUVWXYZ123456]?
```

stop
> Stops the wrap procedure and displays final wrap statistics.

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: `exit`

Chapter 44. Using and Configuring LAN Emulation Clients

This chapter describes LAN Emulation Clients (LECs). It includes the following sections:

- "LAN Emulation Client Overview"
- "Configuring LAN Emulation Clients (LE Client Config>)"
- "Configuring a Specific LAN Emulation Client (LEC Config>)" on page 44-4

LAN Emulation Client Overview

On the router, LECs serve the purpose of "ports" or "interfaces" on traditional routers and bridges. The router bridges and routes traffic between emulated LANs by receiving and transmitting traffic through its LECs.

LEC has two prompt levels:

1. `LE Client Config>` lets you enter commands that control the environment of all your LECs. The commands for this prompt level are described in "Configuring LAN Emulation Clients (LE Client Config>)."

2. One of the commands, **config**, gets you to another prompt level, `LEC Config>`, at which you can enter commands to configure a specific LEC. The commands for this prompt level are described in "Configuring a Specific LAN Emulation Client (LEC Config>)" on page 44-4.

To get to the `LE Client Config>` prompt, enter **le-c** at the ATM Config> prompt as described in "ATM Configuration Commands" on page 42-3.

For information on using, configuring, and monitoring LAN Emulation Services, refer to *Multiprotocol Switched Services (MSS) Server Command Line Interface Volume 1 User's Guide and Protocol Reference,* SC380-3830.

An explanation of commands for LAN Emulation Clients follows.

Configuring LAN Emulation Clients (LE Client Config>)

This section summarizes and explains the commands for configuring and using the set of LE Clients on a particular ATM interface.

Enter the commands at the `LE Client Config>` prompt under the `ATM Config>` prompt, as described in "ATM Configuration Commands" on page 42-3.

Table 44-1 (Page 1 of 2). LAN EMULATION Client Configuration(?) Commands Summary

Command	Function
? (Help)	Lists the console commands or list the actions associated with specific commands
Add	Adds a LEC for the following types of Emulated LANs architectures: • Ethernet • Token Ring

Table 44-1 (Page 2 of 2). LAN EMULATION Client Configuration(?) Commands Summary

Command	Function
Config	Gets you to the `LEC Config>` prompt, from which you can configure a specific LAN Emulation Client as described in: "Configuring a Specific LAN Emulation Client (LEC Config>)" on page 44-4
List	Lists the ATM emulated LANs.
Remove	Removes a LEC.
Exit	Returns to the previous prompt.

Help

Use the **? (help)** command to list the commands that are available from the current prompt level. You also can enter a ? after a specific command name to list its options.

Syntax: ?

Example: ?

```
ADD
CONFIG
LIST
REMOVE
EXIT
```

Add

Use the **add** command to add a LEC for a Token-Ring or Ethernet emulated LAN.

Syntax: add   Ethernet
                Token Ring

token-ring
    Token-ring emulated LAN

Example: add token ring

```
LE Client Config> add token-ring
Added device as interface 3
```

ethernet
    Ethernet emulated LAN

Example: add ethernet

```
LE Client Config> add ethernet
Added device as interface 2
```

Config

Use the **config** command to get you to the `LEC Config>` prompt, from which you can configure the details of a specific LAN Emulation Client. Refer to "Configuring a Specific LAN Emulation Client (LEC Config>)" on page 44-4.

Syntax: <u>co</u>nfig interface#

interface#
> An integer number assigned by the router when the LEC was added to the configuration. Use the **list** command to determine the number inteface number assigned to the LEC.

Example: `config`

```
LE Client Config> config 3
```

List

Use the **list** command to list the LAN emulation clients.

Syntax: <u>l</u>ist

Example: `list`

```
LE Client Config> list
                  ATM Emulated LANs
 --------------------------------------------------------
  ATM interface number = 0
  LEC interface number = 1
  Emulated LAN type    = Token Ring
  Emulated LAN name    =
```

Remove

Use the **remove** command to remove a LEC. You must specify the interface number that was assigned when the LEC was added to the configuration. Use the **list** command to determine the interface number assigned to the LEC.

Syntax: <u>r</u>emove   interface#

interface#
> An integer number assigned by the router.

Exit

Use the **exit** command to return to the previous prompt.

Configuring a Specific LAN Emulation Client (LEC Config>)

This section summarizes and then explains the commands for configuring a specific LAN Emulation Client.

Enter the commands at the LEC Config> prompt after entering the **config** command at the LE Client Config> prompt.

Table 44-2. LAN Emulation Client Configuration Commands Summary	
Command	**Function**
? (Help)	Lists the console commands or lists the actions associated with specific commands
List	Lists the LAN Emulation Client configuration.
Set	Sets the LAN Emulation Client parameters.
Exit	Returns to the previous prompt.

Help

Use the **? (help)** command to list the commands that are available from the current prompt level. You also can enter a ? after a specific command name to list its options.

Syntax: ?

Example: ?

```
LIST
SET
EXIT
```

List

Use the **list** command to list the LE client configuration.

Syntax: list

Example: list

Set

Use the **set** command to set LE Client parameters.

Syntax: set   aging-time
               arp-cache-size
               arp-queue-depth
               arp-response-time
               auto-config
               best-effort-peakrate
               config-retries
               conn-completion-time
               control-timeout
               elan-name
               esi-address
               flush-timeout
               forward-delay
               frame-size

lecs-atm-address
mac-address
les-atm-address
multicast-send-avg
multicast-send-peak
multicast-send-type
path-switch-delay
retry-count
selector
trace
unknown-count
unknown-time
vcc-timeout

aging-time

Sets aging time. This is the maximum time that a LEC will maintain an entry in its LE_ARP cache in the absence of a verification of that relationship. A larger aging time may result in a faster session setup time, but may also use more memory and reacts slower to changes in the network.

Valid Values: An integer number of seconds in the range of 10 to 300.

Default Value: 300

Example: `set aging-time`

  LEC Config> `set aging-time 200`

arp-cache-size

Sets the number of entries in the ARP cache.

The size of the ARP cache limits the number of simultaneous data direct VCCs. Larger ARP caches require more memory, but permits the client to simultaneously converse with a larger number of destinations.

Valid Values: An integer number in the range of 10 to 1024.

Default Value: 10

Example: `set arp-cache-size`

  LEC Config> `set arp-cache-size 10`

arp-queue-depth

Sets the maximum number of queued frames per ARP cache entry.

The LEC enqueues frames when switching the data path from the Multicast Send VCC to a Data Direct VCC. Frames passed to the LEC for transmission will be discarded if the queue is full. A larger frame queue requires more memory, but results in fewer discarded frames during the data path switch.

Valid Values: An integer number in the range of 0 to 10.

Default Value: 5

Example: `set arp-queue-depth`

  LEC Config> `set arp-queue-depth 10`

arp-response-time

Sets expected ARP response time.

This value controls how frequently an unanswered LE ARP request is

retried. Larger values result in fewer LE ARPs, which causes less traffic and possibly increases the amount of time before a Data Direct VCC is established.

Valid Values: An integer number of seconds in the range of 1 to 30
Default Value: 1 second

Example: `set arp-response-time`

> LEC Config> `set arp-response-time 20`

auto-config

Specifies whether this LEC uses LECS auto-config mode. Specify YES or NO. The LEC may contact the LECS to obtain the address of its LES and various other configuration parameters. This value must be Yes for the client to use a redundant IBM MSS Server LES.

Valid Values: If YES, then you do not have to configure the ATM address of the LES.

If NO, then you *must* configure the ATM address of the LES using the **set les-atm-address** command as described on page 44-9.
Default Value: NO

Example: `set auto-config`

> LEC Config> `set auto-config yes`

best-effort-peakrate

Sets the Best Effort Peak Rate. Used when establishing best effort multicast send connections.

The maximum peak rate depends on the maximum data rate of the ATM device.

Specify an integer from 10 to the maximum peak rate in Kbps (the definition is the maximum data rate) as follows:

- If ATM maximum data rate is 25 Mbps, the maximum peak rate is 25,000 Kbps.
- If ATM maximum data rate is 155 Mbps, the maximum peak rate is 155,000 Kbps.

Mismatched PCRs may result in poor performance due to excessive retransmissions. An example would be when a 155 Mbps LEC connects with a 25 Mbps client. If the LEC transmits at its maximum data rate, then the other client can be easily overrun with data. In situations where the LEC communicates with many clients with significantly less capacity, performance may be improved by lowering this value.

Valid Values: An integer number in the range of 10 - device maximum data rate.
Default Value: 155000 Kbps

Example: `set best-effort-peakrate`

> LEC Config> `set best-effort-peakrate 24000`

config-retries

Sets the number of configuration retries. This value specifies how many times the client will retry the configuration phase of joining an ELAN (in the event of failure.)

Specify an integer number in the range 0 to 5. If the client is unable to connect to the LES address given to it in a configure response, the client will send another configure request to the LECS before releasing the configured Data Direct VCC. This value controls how many retries are made before the client gives up. When this limit is reached, the client will sleep for a period of time (not exceeding 30 seconds) and restart the configuration phase. Interoperability problems may arise if this value is set to zero.

Valid Values: An integer number in the range of 0 to 5.
Default Value: 3

Example: `set config-retries`

   LEC Config> **`set config-retries 4`**

connection-completion-time

Sets the connection completion time. This is the time interval in which data or a READY_IND message is expected from a calling party.

Specify an integer number of seconds in the range 1 to 10. When a Data Direct VCC is established to the client, the LEC expects data or a READY_IND message within this time period. The LEC will not transmit frames over a Data Direct VCC established to it until receiving data or a READY_IND. This parameter value controls the amount of time which passes before the LEC issues a READY QUERY (in hopes of receiving a READY_IND). Smaller values lead to faster response times, but also to unnecessary transmissions.

Valid Values: An integer number of seconds in the range of 1 to 10.
Default Value: 4 seconds

Example: `set connection-completion-time`

   LEC Config> **`set connection-completion-time 5`**

control-timeout

Sets the control timeout. This is the timeout period used for timing out most request/response control frame interactions. Smaller timeout values result in more traffic and may result in faster response times.

Valid Values: An integer number of seconds in the range of 10 to 300.
Default Value: 30 seconds

Example: `set control-timeout`

   LEC Config> **`set control-timeout 100`**

elan-name

Specifies name of the ELAN that the LEC wishes to join. This is the ELAN name sent to the LECS in the configure request (if the LEC autoconfigures) or to the LES in the join request. The LECS or LES may return a different ELAN name in the response.

Valid Values: Any character string length of 0 - 32 bytes.
Default Value: Blank

Note: A blank name (0 length string) is valid.

Example: `set elan-name`

   LEC Config> **`set elan-name FUZZY`**

esi-address

Sets the ESI portion of the LEC's ATM address.

Specify the ESI portion (octets 13 through 19) of the LEC's ATM address. The ESI and selector combination of the LEC must be unique among all LAN emulation components on the MSS Server.

Valid Values: Any 12 hexadecimal digits.
Default Value: none

Example: `token ring lec config>set esi`

```
Select ESI
   (1) Use burned in ESI
   (2) 11.22.33.44.55.66

Enter selection [1]?
```

flush-timeout

Sets the flush timeout. This is the time limit to wait to receive the LE_FLUSH_RESPONSE after the LE_FLUSH_REQUEST has been sent before taking recovery action. When switching from the multicast send to a data direct data path, the client sends a flush request over the multicast send VCC. Until a flush response is received, or until the path switch delay expires, frames are queued for the destination. A smaller timeout generates more flush traffic, but may result in fewer discarded data frames.

Valid Values: An integer number of seconds in the range of 1 to 4.
Default Value: 4 seconds

Example: `set flush-timeout`

```
LEC Config> set flush-timeout 3
```

forward-delay

Sets the forward delay. Entries in the LE ARP cache must be periodically reverified. The forward delay time is the maximum amount of time a remote entry may remain in the cache during a network topology change. Larger aging times may result in stale (invalid) entries, but also cause less reverification traffic.

Valid Values: An integer number of seconds in the range of 4 to 30.
Default Value: 15 seconds

Example: `set forward-delay`

```
LEC Config> set forward-delay 10
```

frame-size

Sets the frame size.

The value specified for frame-size must be equal to or less than the value specified for ATM max-frame using the ATM INTERFACE> **set max-frame** command as described on page 42-7.

The value specified for frame-size must be equal to or less than the value specified for ATM max-frame using the ATM INTERFACE> **set max-frame** command as described on page 42-7. The specified maximum frame size includes any Layer 2 header information. All transparently bridged LANs must have the same maximum frame size.

Valid Values:
                    1516
                    4544
                    9234
                    18190
Default Value: If the ELAN type is token ring, the default is 4544. If
                      the ELAN type is Ethernet, the default is 1516.

Example: `set frame-size`

```
LEC Config> set frame-size 4544
```

lecs-atm-address
          Specifies the ATM address of the corresponding LECS.

          If the client is set to auto configure, it attempts to connect to a LECS. If
          it is unable to connect to a LECS, then it may try another LECS ATM
          address. The LECS ATM addresses that are tried, in order, are:

          1. This configured LECS address
          2. Any LECS address obtained through ILMI
          3. The well-known LECS address defined by the ATM Forum.

          Note: This command should be entered on one command line. It is
          shown here on two lines because of spacing.

          Example: `set lecs-atm-address`

```
LEC Config> set lecs-atm-address
     39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.01
```

les-atm-address
          Sets the LES ATM address. This command may be optional or required
          depending upon the setting of lecs-auto-config as described in the **set
          lecs-auto-config** command on page 44-6.

          • If lecs-auto-config is YES, the les-atm-address may not be
            configured.
          • If lecs-auto-config is NO, then the les-atm-address is required.

          Specify the ATM address of the LES.

          Note: This command should be entered on one command line. It is
          shown here on two lines because of spacing.

          Example: `set les-atm-address`

```
LEC Config> set les-atm-address
     39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.02
```

mac-address
          Sets the MAC address for this LE client. You *may* specify that the client
          use the burned in MAC address, or you may specify a different MAC
          address. If you have two clients that are bridged together, then they
          both cannot use the same MAC address.

          This MAC address is registered with the LES when the client joins the
          ELAN.

          Valid Values: Any valid MAC address.
          Default Value: none

          Example: `set mac-address`

```
LEC Config> set mac-address FF.FF.FF.FF.FF.FF.01
```

multicast-send-avg

> Sets the multicast send VCC average rate in Kbps. Used by the LEC for reserving bandwidth on the VCC to the BUS. It specifies the forward and backward sustained cell rate used when setting up a reserved bandwidth multicast send VCC.
>
> This parameter is only applicable when the multicast-send-type is variable or constant. If multicast-send-avg equals multicast-send-peak, then a constant bit rate (CBR) multicast send is signalled. Otherwise, a variable bit rate (VBR) musticast send is signalled. Multicast-send-avg must at least equal multicast-send peak.
>
> A variable or constant multicast send VCC may improve data transfer rates in congested networks, but reserving bandwidth and not using it wastes network resources.
>
> When the multicast-send-type is variable or constant, then multicast-send-avg and multicast-send-peak must be specified.
>
> **Example:** `set multicast-send-avg`
>
> > `LEC Config> set multicast-send-avg 4000`

multicast-send-peak

> Sets the multicast send peak rate in Kbps. Used by LEC for reserving bandwidth on the VCC to the BUS. It specifies the forward and backward peak cell rate used when establishing a reserved bandwidth multicast send VCC.
>
> This parameter is only applicable when the multicast-send-type is reserved bandwith. If multicast-send-avg equals multicast-send-peak, then a constant bit rate (CBR) multicast send is signalled. Otherwise, a variable bit rate (VBR) musticast send is signalled. Multicast-send-avg must at least equal multicast-send peak.
>
> A reserved bandwidth multicast send VCC may improve data transfer rates in congested networks, but reserving bandwidth and not using it wastes network resources.
>
> When the multicast-send-type is variable or constant, then multicast-send-avg and multicast-send-peak must be specified.
>
> **Example:** `set multicast-send-peak`
>
> > `LEC Config> set multicast-send-peak 155`

multicast-send-type

> Sets the multicast send type. Specifies the method used by the LEC when establishing the multicast send VCC.
>
> If multicast-send-avg equals multicast-send-peak, then a constant bit rate (CBR) multicast send is signalled. Otherwise, a variable bit rate (VBR) multicast send is signalled. Multicast-send-avg must at least equal multicast-send peak.
>
> A reserved bandwidth multicast send VCC may improve data transfer rates in congested networks, but reserving bandwidth and not using it wastes network resources.
>
> When the multicast-send-type is variable or constant, then multicast-send-no and multicast-send-peak must be specified.
>
> **Valid Values:**   Best Effort or Variable or Constant

> **Default Value:**   Best Effort
>
> **Note:** These traffic parameters are similar to those for LES-BUS (LAN Emulation Services - Broadcast and Unknown Server). If you are using LAN Emulation Services, refer to *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 1 Release 1* SC380-3830.
>
> **Example:** `set multicast-send-type`
>
> > LEC Config> `set multicast-send-type best-effort`

path-switch-delay

> Sets the path switch delay.
>
> The LEC must ensure that all frames sent through the BUS to a destination have arrived at the destination before it can start using a Data Direct VCC. This is accomplished using the flush protocol, or by waiting path-switch-delay seconds after sending the last packet to the BUS. Smaller values improve performance, but may result in out-of-order packets in a heavily congested network.
>
> **Valid Values:**   An integer number of seconds in the range of 1 to 8.
> **Default Value:**   6 seconds
>
> **Example:** `set path-switch-delay`
>
> > LEC Config> `set path-switch-delay 5`

retry-count

> Sets the retry count. This is maximum number of times that the LEC retries an LE_ARP_REQUEST for a specific frame's LAN destination. If no ARP response is received after the specified number of retries, then the entry is purged from the LE ARP cache.
>
> **Valid Values:**   An integer number in the range of 0 to 2.
> **Default Value:**   1
>
> **Example:** `set retry-count`
>
> > LEC Config> `set retry-count 2`

selector

> Specifies a value to use as Selector field component of local ATM address used when calling the LECS.
>
> The combination of ESI and selector must be unique among all LAN Emulation entries on the MSS Server.
>
> By default a selector is selected for the configured ESI.
>
> **Valid Values:**   Any octet in hexadecimal, 00 to FF, that has not already been assigned to the LECS ATM device. The combination of LECS ESI and LECS selector must be unique among all LAN Emulation components using the LECS ATM device.
>
> **Example:** `set selector`
>
> > LEC Config> `set selector 01`

trace

> Enables packet tracing for the LEC. To perform packet tracing, three steps are required:
>
> 1. Enable packet tracing system (under ELS)

2. Enable tracing on the LEC subsystem (under ELS)

3. Enable packet tracing on the desired LECs (using this command).

Valid Values: Enable or Disable
Default Value: Disable

Example: `Token Ring LEC config>`**`set trace`**

  `Trace packets on the LEC? [No]?`**`yes`**

unknown-count

Sets the unknown frame count. This is the maximum number of frames for a specific unicast MAC address or route descriptor that may be sent to the BUS within the time specified by the unknown-time parameter.

Larger values decrease the number of discarded frames while increasing the load on the BUS.

Valid Values: An integer number of frames in the range of 1 to 255.
Default Value: 10

Example: **`set unknown-count`**

  `LEC Config>` **`set unknown-count`**

unknown-time

Sets the unknown frame time. This is the time interval during which the maximum number of frames for a specific unicast MAC address or route descriptor (specified by the unknown-count parameter) may be sent to the BUS. Larger values increase the number of discarded frames while decreasing the load on the BUS.

Valid Values: An integer number of seconds in the range of 1 to 60.
Default Value: 1

Example: **`set unknown-time`**

  `LEC Config>` **`set unknown-time 5`**

vcc-timeout

Sets the VCC timeout. Data direct VCCs that are idle this long should be released.

Valid Values: An integer number of seconds in the range of 1 to 31536000.
Default Value: 1200

Note: This parameter is meaningful only for SVC connections.

Example: **`set vcc-timeout`**

  `LEC Config>` **`set vcc-timeout 1000`**

Exit

Use the **exit** command to return to the `LE Client Config>` prompt.

Chapter 45. Monitoring LAN Emulation Clients

This chapter describes how to monitor LAN Emulation Clients (LECs). It includes the following sections:

- "Accessing the LEC Console Environment"
- "LEC Console Commands" on page 45-2

Accessing the LEC Console Environment

Use the following procedure to access the LEC console commands. This process gives you access to the LEC *monitoring* process.

1. At the OPCON prompt, enter **talk 5**. (For more detail on this command, refer to Chapter 2, "The OPCON Process and Commands.") For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the + prompt, enter the **network** command to display the network interface numbers for which the router is currently configured, and enter the *interface number* for the LEC you wish to monitor. For example:

   Example: network

   ```
    + network

    1  : ATM Ethernet LAN Emulation: ETH
    2  : IP Protocol Network
    3  : Bridge Application
    5  : CHARM ATM Adapter
   Network number [0]? 1
   LEC+
   ```

   The LEC monitoring prompt (LEC+), is displayed.

   If you know the interface number of the LEC you wish to monitor, enter the **network** command followed by the *interface number* of the LEC.

   ```
    + network 1
   LEC+
   ```

LEC Console Commands

This section summarizes and then explains the LEC console commands. You can access LEC console commands at the `LEC+` prompt. Table 45-1 shows the commands.

Table 45-1. LE Config Console Command Summary	
Command	**Function**
? (Help)	Lists the LEC console commands or lists the options associated with specific commands.
List	Lists: • LEC Address Resolution Table (ARP) • LEC configuration • Data Direct VCC information • LEC statistics.
MIB	Displays LEC MIB objects including: • LEC MIB Configuration Table • LEC MAC ARP Table • LEC Route Descriptor Table • LEC MIB Server VCC Tables • LEC MIB Statistics Table • LEC MIB Status Table
Exit	Exits the LEC console process.

? (Help)

Use the **? (help)** command to list the commands that are available from the current prompt level. You can also enter a ? after a specific command name to list its options.

Syntax: ?

Example: ?

List

Use the **list** command to list the LEC Address Resolution Table (ART), list the LEC configuration, list Data Direct VCC information, or list LEC statistics.

Syntax: <u>list</u>   <u>arp</u>
               <u>configuration</u>
               <u>data</u>
               <u>statistics</u>

arp
   Lists the LEC Address Resolution Table (ART).

   Example: list arp

```
LEC+ list arp

          LEC Address Resolution Table

Max Table Size          = 10        Total # of entries
Free Table Entries      = 10        # of free entries.
Current Mac Entries     = 0         MAC - ATM  entries
Current RD Entries      = 0         RD - ATM entries
Arp Aging Time          = 300       Time for an entry to be 'aged' out
Verify Sweep Interval   = 60        Table updated/verified every 60 seconds
```

configuration

> Lists the LEC configuration.

For Ethernet:

Example: `list config`

```
LEC+ list config

          ATM LEC Configuration

ATM interface number            = 0
LEC interface number            = 1
LECS auto configuration         = No

C1: Primary ATM address
        ESI address             = Use burned in addr
        Selector byte           = 3
C2: Emulated LAN type           = Ethernet
C3: Maximum frame size          = 1516
C5: Emulated LAN name           =
C6: LE Client MAC address       = Use burned in addr
C7: Control timeout             = 120
C9: LE Server ATM address       = 39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.02
C10: Maximum unknown count       = 1
C11: Maximum unknown time        = 1
C12: VCC timeout period          = 1200
C13: Maximum retry count         = 1
C17: Aging time                  = 300
C18: Forward delay time          = 15
C20: LE ARP response time        = 1
C21: Flush timeout               = 4
C22: Path switch delay           = 6
C24: Multicast send VCC type     = Best-Effort
C25: Multicast send VCC avg rate = 25000000
C26: Multicast send VCC peak rate = 25000000
C28: Connection completion timer = 4

LE ARP queue depth              = 10
LE ARP cache size               = 10
Best effort peak rate           = 25000
Maximum config retries          = 3
```

For Token Ring:

Example: `list config`

```
LEC+list config

                    ATM LEC Configuration

  ATM interface number          = 0
  LEC interface number          = 1
  LECS auto configuration       = No

  C1: Primary ATM address
         ESI address            = Use burned in addr
         Selector byte          = 2
  C2: Emulated LAN type         = Token Ring
  C3: Maximum frame size        = 4544
  C5: Emulated LAN name         =
  C6: LE Client MAC address     = 10.00.5A.11.11.11
  C7: Control timeout           = 120
  C9: LE Server ATM address     = 39.84.0F.00.00.00.00.00.00.00.00.00.01.10.00.5A.00.DE.AD.02
  C10: Maximum unknown count    = 1
  C11: Maximum unknown time     = 1
  C12: VCC timeout period       = 1200
  C13: Maximum retry count      = 1
  C17: Aging time               = 300
  C18: Forward delay time       = 15
  C20: LE ARP response time     = 1
  C21: Flush timeout            = 4
  C22: Path switch delay        = 6
  C24: Multicast send VCC type      = Best-Effort
  C25: Multicast send VCC avg rate  = 25000
  C26: Multicast send VCC peak rate = 25000
  C28: Connection completion timer  = 4

  LE ARP queue depth            = 10
  LE ARP cache size             = 36920
  Best effort peak rate         = 25000
  Maximum config retries        = 999
  Packet trace                  = Yes
  RIF Aging Timer               = 120
  Source Routing                = Enabled
  IPX interface configuration record missing
```

data

Lists the LEC Data Direct VCC information.

Example: `list data`

```
LEC+ list data

        LEC Data Direct VCC Table

  Max Table Size    = 20      Max no of SVC connections
  Current Size      = 0       Currently used
  Max Conn Handles  = 512
  Inactivity Timeout = 1200   No Data Xfer Timeout before connection is
                              closed (seconds)
  Sweep Interval    = 60      Time interval for activity check
```

statistics

Lists LEC statistics.

Example: `list statistics`

```
LEC+ list stat

        LEC Statistics

  In Octets.high   = 0       No of Bytes received
  In Octets.low    = 346
  In Discards      = 2       Packets discarded
  In Errors        = 0       Rx.Errors
  In Unknown Protos = 0      Unknown protocols received
  Out Octets.high  = 0       No of Bytes xmitted.
  Out Octets.low   = 0
  Out Discards     = 0
  Out Errors       = 0       Tx.Errors
```

MIB

Use the **mib** command to display MIB objects.

Note: Some of this information may be displayed in a different format using the **list** command.

Syntax: <u>mib</u>   <u>config</u>
          <u>mac</u>
          <u>rd</u>
          <u>server</u>
          <u>statistics</u>
          <u>status</u>
          <u>trace</u>

config

Displays the LEC MIB Configuration Table.

Example: `list mib config`

```
LEC+ mib config

lecConfigTable:
lecConfigMode               = Manual
lecConfigLanType            = 802.3 - Ethernet
lecConfigMaxDataFrameSize   = 1516
lecConfigLanName            =
lecConfigLesAtmAddress      = 39.84.0F.00.00.00.00.
lecControlTimeout           = 120
lecMaxUnknownFrameCount     = 1
lecMaxUnknownFrameTime      = 0
lecVccTimeoutPeriod         = 1200
lecMaxRetryCount            = 1
lecAgingTime                = 300
lecForwardDelayTime         = 15
lecExpectedArpResponseTime  = 1
lecFlushTimeout             = 4
lecPathSwitchingDelay       = 6
lecLocalSegmentId           = 0
lecMulticastSendType        = 1
lecMulticastSendAvgRate     = 25000000
lecMulticastSendPeakRate    = 25000000
lecConnectionCompleteTimer  = 4
```

lecConfigMode	LEC config mode: AUTO or MANUAL. If AUTO, LEC Uses LECS to get the LES ATM address.
lecConfigLanType	LAN type, either Ethernet or token-ring
lecConfigMaxDataFrameSize	Maximum frame size
lecConfigLanName	ELAN Name
lecConfigLesAtmAddress	LE Server ATM address
lecControlTimeout	Timeout for request/response control frame
lecMaxUnknownFrameCount	Maximum bumber of unknown frames
lecMaxUnknownFrameTime	Period in which LEC will send a maximum of MaxUnknownFrameCount frames to the BUS for a given unicast LAN Destination, and it must also initiate the address

	resolution protocol to resolve that LAN Destination.
lecVccTimeoutPeriod	Inactivity timeout of SVC Data Direct VCCs
lecMaxRetryCount	LE ARP retry count
lecAgingTime	Life of unverified entry in the ARP table
lecForwardDelayTime	
lecExpectedArpResponseTime	ARP Request/Response cycle time
lecFlushTimeout	LE Flush Request/Flush Reply timeout period
lecPathSwitchingDelay	
lecLocalSegmentId	Segment ID of emulated LAN.Only for 802.5 clients
lecMulticastSendType	Signaling parameter used by LEC for multicast send VCC
lecMulticastSendAvgRate	Signaling parameter used by LEC for multicast send VCC
lecMulticastSendPeakRate	Signaling parameter used by LEC for multicast send VCC
lecConnectionCompleteTimer	

mac

Displays the LEC MAC ARP Table

rd

Displays the LEC Route Descriptor Table

server

Displays the LEC MIB Server VCC Tables

Example: `mib server`

```
LEC+ mib server

lecServerVccTable:
    lecConfigDirectInterface      = 0
    lecConfigDirectVpi            = 0
    lecConfigDirectVci            = 0
    lecControlDirectInterface     = 1
    lecControlDirectVpi           = 0
    lecControlDirectVci           = 38
    lecControlDistributeInterface = 1
    lecControlDistributeVpi       = 0
    lecControlDistributeVci       = 37
    lecMulticastSendInterface     = 1
    lecMulticastSendVpi           = 0
    lecMulticastSendVci           = 34
    lecMulticastForwardInterface  = 1
    lecMulticastForwardVpi        = 0
    lecMulticastForwardVci        = 33
```

lecConfigDirectInterface	The interface associated with the Configuration Direct VCC
lecConfigDirectVpi	VPI which identifies the above VCC if it exists

lecConfigDirectVci	VCI which identifies the above VCC if it exists
lecControlDirectInterface	The interface associated with the Control Direct VCC
lecControlDirectVpi	VPI which identifies the above VCC if it exists
lecControlDirectVci	VCI which identifies the above VCC if it exists
lecControlDistributeInterface	The interface associated with the Control Distribute VCC
lecControlDistributeVpi	VPI which identifies the above VCC if it exists
lecControlDistributeVci	VCI which identifies the above VCC if it exists
lecMulticastSendInterface	The interface associated with the Multicast Send VCC
lecMulticastSendVpi	VPI which identifies the above VCC if it exists
lecMulticastSendVci	VCI which identifies the above VCC if it exists
lecMulticastForwardInterface	The interface associated with the Multicast Forward VCC
lecMulticastForwardVpi	VPI which identifies the above VCC if it exists
lecMulticastForwardVci	VCI which identifies the above VCC if it exists

statistics
    Displays the LEC MIB Statistics Table.

Example: `mib statistics`

```
LEC+ mib statistics

lecStatisticsTable:
    lecArpRequestsOut        = 1
    lecArpRequestsIn         = 0
    lecArpRepliesOut         = 0
    lecArpRepliesIn          = 1
    lecControlFramesOut      = 2
    lecControlFramesIn       = 2
    lecSvcFailures           = 1

    In Frames                = 919
    Out Frames               = 917
    In Bytes                 = 82698
    Out Bytes                = 82482
```

lecArpRequestsOut	No. of LE ARP requests sent by this LEC
lecArpRequestsIn	No. of LE ARP requests received by this LEC
lecArpRepliesOut	No. of LE ARP responses sent by this LEC

lecArpRepliesIn	No. of LE ARP responses received by this LEC
lecControlFramesOut	No. of Control Packets sent by this LEC
lecControlFramesIn	No. of Control Packets received by this LEC
lecSvcFailures	The total number of:

- Outgoing LAN Emulation SVCs which this client tried but failed, to open
- Incoming LAN Emulation SVCs which this client tried, but failed to establish
- Incoming LAN Emulation SVCs which this client rejected for protocol or security reasons

status

Lists MIB status.

Example: `list mib status`

```
LEC+ mib status

lecStatusTable:
  lecPrimaryAtmAddress        = 39.84.0F.00.00.00
  Client ATM address=         = 00.00.00.00.00.01.10.00.5A.00.DE.AD.03
  lecId                       = 1                   Assigned by LES
  lecInterfaceState           = Operational         State of the LEC
  lecLastFailureRespCode      = None                Error code from last
                                                    failed Config/Join resp.
  lecLastFailureState         = Initial State       State of LEC when
                                                    updating above field.
  lecProtocol                 = 1                   Protocol specified by
                                                    LEC in Join requests.
  LecVersion                  = 1                   LEC Protocol Version
                                                    of above
  lecTopologyChange           = False
  lecConfigServerAtmAddress   = 00.00.00.00.00.00.
  ATM Address of LECS         = 00.00.00.00.00.00.00.00.00.00.00.00.00
  lecConfigSource             = Did not use LECS
  lecActualLanType            = 802.3 - Ethernet    Frame format currently
                                                    used by LEC
  lecActualMaxDataFrameSize   = 1516
  lecActualLanName            = ETH                 Name of emulated LAN
                                                    that LEC joined.
  lecActualLesAtmAddress      = 39.84.0F.00.00.00.
  ATM Address of LES          = 00.00.00.00.00.00.01.10.00.5A.00.DE.AD.02
  lecProxyClient              = False               Is LES acting like a
                                                    proxy ?
```

trace

Example: trace on or **off** or **list**

Exit

Use the **exit** command to return to the previous prompt level.

Syntax: e̲xit

Example: exit

Appendix A. X.25 National Personalities

This appendix lists the default settings for GTE-Telenet and DDN.

GTE-Telenet

The following parameters are the default settings for GTE-Telenet:

- Callreq: 20
- Clearreq:
  - Retries: 1
  - Timer: 18
- Disconnect: Passive
- DP-timer: 500 milliseconds
- Frame window size: 7
- Network Type: CCITT
- N2 timeouts: 20
- Packet:
  - Default size: 128
  - Maximum size: 256
  - Window size: 2
- Reset
  - Retries: 1
  - Timer: 18
- Restart
  - Retries: 1
  - Timer: 18
- Standard: 1984
- T1-timer: 4
- T2-timer: 2

DDN

The following parameters are the default settings for DDN:

- Callreq: 20
- Clearreq:
  - Retries: 1
  - Timer: 18
- Disconnect: Passive
- DP-timer: 500 milliseconds
- Frame window size: 7
- Network Type: CCITT
- N2 timeouts: 20
- Packet:
  - Default size: 128
  - Maximum size: 256
  - Window size: 2
- Reset
  - Retries: 1
  - Timer: 18
- Restart
  - Retries: 1
  - Timer: 18
- Standard: 1984
- T1-timer: 4
- T2-timer: 2

Appendix B. Making a Router Load File from Multiple Disks

If a software load arrives on multiple disks, use the procedure in the following
sections to combine the loads into one load file that the router can use at the time
of booting.

The first disk contains the following four files that you need if you want to fragment
an existing load for transport on multiple diskettes.

cutup.c　　　(UNIX C source file that can be compiled using a standard C
　　　　　　　compiler)

cutup.exe　　(DOS)

Use the following files for reassembling the load fragments onto a DOS or UNIX
server.

kopy.bat　　(DOS)

kopy　　　　(UNIX shell script)

Assembling a Load File Under DOS

To assemble a load from the two diskettes, use the DOS batch file provided on
diskette 1 (KOPY.BAT) using the following syntax:

kopy`<installation_drive><destination_directory>`

Before assembling the load make sure that you have created a destination
directory, and that you have inserted the first diskette in the drive specified by the
installation_diskette_drive parameter. The following example illustrates this
procedure.

```
B:\>kopy b: c:\source\cutup\tmp
B:\>copy c:\gw0/B c:\source\cutup\tmp\gw.tmp
1 file(s) copied
.
Please mount the second diskette
Press any key to continue . . .
Copying the second load file fragment
B:\>
B:\>copy c:\source\cutup\tmp\gw.tmp/B + b:\gw1
c:\source\cutup\tmp\gw.tmp  c:\SOURCE\CUTUP\TMP\GW.TMP
B:\GW1
1 file(s) copied
B:\>rename c:\source\cutup\tmp\gw.tmp  gw.ldc
Load file reassembly was successful
B:>
```

Assembling a Load File Under UNIX

To assemble a load from two UNIX diskettes, you can use the UNIX Bourne shell
script (kopy) provided on diskette 1 using the following syntax:

kopy`<installation_drive><diskette_directory><destination_directory`

Before assembling the load make sure that you have created the mount and
destination directories, and that you have inserted the first diskette in the drive
specified by the installation_diskette_drive parameter. The following example
illustrates this procedure.

```
kopy /dev/fd0 /kew /pcfs

Please insert the first diskette

Copying the first load file fragment

Please mount the second diskette

Copying the second load file fragment

Load file reassembly was successful

# ls /kew

gw0   gw1   gw.ldc
```

If you can't use the UNIX Bourne shell script, you can assemble the load manually using the following procedure:

1. Copy the load fragments on the two diskettes (gw0 and gw1) into a directory on the UNIX file system.

2. Type the following UNIX command:

   ```
   cat gw0 gw1 > gw.ldc
   ```

   The resulting file (gw.ldc) is the assembled router load.

Disassembling a Load File Under DOS

To disassemble a load under DOS, use the CUTUP.EXE file as follows:

cutup<file_extension><file_name><cut_length>

The file_extension is attached to the front of each slice needed to cut. The file_name is the DOS file name of the file to be disassembled. The cut_length is the length that CUTUP.EXE makes each fragment as it disassembles the file. The following example illustrates this procedure.

```
C: \source\cutup>dir
Volume in drive C has no label
Volume Serial Number is XXXXXXXX
Directory od C: \SOURCE\CUTUP
.0730934:46p
..0730934:46p
GW      LDC 10225660728931:22p
CUTUP   EXE   105410902939:38a
2 file(s)   1033107 bytes
14811136 bytes free
C: \source\cutup>cutup gw.ldc gw 1000000
.......................................................
....................
............................
c: \SOURCE\CUTUP>dir
Volume in drive C has no label
Volume Serial Number is XXXXXXXX
Directory od C: \SOURCE\CUTUP
.0730934:46p
..0730934:46p
GW      0 10000000801931:22p
GW      LDC    10225660728931:22p
CUTUP   EXE   105410902939:38a
GW      1  225660801931:22p
4 file(s)   2055673 bytes
14811136 bytes free
```

Disassembling a Load File Under UNIX

To disassemble a load under use cutup.c. Begin by compiling the program using your UNIX compiler to make a cutup executable file. Then use the following syntax:

cutup`<file_extension><file_name><cut_length>`

The file_extension is attached to the front of each slice needed to cut. The file_name is the DOS file name of the file to be disassembled. The cut_length is the length CUTUP.EXE that is used to disassemble the file. The following example illustrates this procedure.

```
#  ls -la
total 658
drwxrxrx    2 root      512 Aug   114:41  .
drwxrxrx   26 root     1024 Aug   114:41  ..
drwxrxrx    2 root    24576 Aug   114:41  cutup
drwxrxrr    2 root1022566 Aug   114:41  gw.ldc

#  cutup gw.ldc gw 100000

#  ls -la

total 658
drwxrxrx    2 root      512 Aug   114:41  .
drwxrxrx   26 root     1024 Aug   114:41  ..
drwxrxrx    2 root    24576 Aug   114:41  cutup
drwxrxrr    2 root1022566 Aug   114:41  gw.ldc
drwxrxrr    2 root1000000 Aug   114:41  gw0
drwxrxrr    2 root    22566 Aug   114:41  gw1
```

Appendix C. Licensed Program Materials Availability

This program, 5765-B87, is licensed under the IBM Customer Agreement in the U.S., Canada and Asia Pacific countries and under the International Program License Agreement in Europe and Latin American countries.

Supplemental Terms

Testing Period

None

Installation/Location License

Not applicable. A separate license is required for each machine on which the license program will be used.

Usage License

Not applicable.

Type/Duration of Program Services

Central Service, including the IBM Support Center, will be available until discontinued by IBM with a minimum of six months written notice.

Warranty

IBM warrants that:

1. IBM has the right to license this program.
2. The IBM program conforms to its specifications.

The warranty period for this program expires when its program services are no longer available. During the warranty period, IBM will provide warranty service, without charge, through Program Services. Program Services are available for a warranty program for at least one year following the program's general availability.

Additional Information

Any other documentation with respect to this licensed program, including any documentation referenced herein, is provided for reference purposes only and does not extend or modify these specifications.

List of Abbreviations

AARP	AppleTalk Address Resolution Protocol
ABR	area border router
ack	acknowledgement
AIX	Advanced Interactive Executive
AMA	arbitrary MAC addressing
AMP	active monitor present
ANSI	American National Standards Institute
AP2	AppleTalk Phase 2
APPN	Advanced Peer-to-Peer Networking
ARE	all-routes explorer
ARI/FCI	address recognized indicator/frame copied indicator
ARP	Address Resolution Protocol
AS	autonomous system
ASBR	autonomous system boundary router
ASCII	American National Standard Code for Information Interchange
ASN.1	abstract syntax notation 1
ASRT	adaptive source routing transparent
ASYNC	asynchronous
ATCP	AppleTalk Control Protocol
ATP	AppleTalk Transaction Protocol
AUI	attachment unit interface
ayt	are you there
BAN	Boundary Access Node
BECN	backward explicit congestion notification
BGP	Border Gateway Protocol
BNC	bayonet Niell-Concelman
BNCP	Bridging Network Control Protocol
BOOTP	BOOT protocol
BPDU	bridge protocol data unit
bps	bits per second
BR	bridging/routing
BRS	bandwidth reservation
BSD	Berkeley software distribution
BTP	BOOTP relay agent
BTU	basic transmission unit
CAM	content-addressable memory

CCITT	Consultative Committee on International Telegraph and Telephone
CD	collision detection
CGWCON	Gateway Console
CIDR	Classless Inter-Domain Routing
CIR	committed information rate
CLNP	Connectionless-Mode Network Protocol
CPU	central processing unit
CRC	cyclic redundancy check
CRS	configuration report server
CTS	clear to send
CUD	call user data
DAF	destination address filtering
DB	database
DBsum	database summary
DCD	data channel received line signal detector
DCE	data circuit-terminating equipment
DDLC	dual data-link controller
DDN	Defense Data Network
DDP	Datagram Delivery Protocol
DDT	Dynamic Debugging Tool
DHCP	Dynamic Host Configuration Protocol
dir	directly connected
DL	data link
DLC	data link control
DLCI	data link connection identifier
DLS	data link switching
DLSw	data link switching
DMA	direct memory access
DNA	Digital Network Architecture
DNCP	DECnet Protocol Control Protocol
DNIC	Data Network Identifier Code
DoD	Department of Defense
DOS	Disk Operating System
DR	designated router
DRAM	Dynamic Random Access Memory
DSAP	destination service access point
DSE	data switching equipment
DSE	data switching exchange

DSR	data set ready		**IGP**	interior gateway protocol
DSU	data service unit		**InARP**	Inverse Address Resolution Protocol
DTE	data terminal equipment		**IP**	Internet Protocol
DTR	data terminal ready		**IPCP**	IP Control Protocol
Dtype	destination type		**IPPN**	IP Protocol Network
DVMRP	Distance Vector Multicast Routing Protocol		**IPX**	Internetwork Packet Exchange
E1	2.048 Mbps transmission rate		**IPXCP**	IPX Control Protocol
EDEL	end delimiter		**ISDN**	integrated services digital network
EDI	error detected indicator		**ISO**	International Organization for Standardization
EGP	Exterior Gateway Protocol		**Kbps**	kilobits per second
EIA	Electronics Industries Association		**LAN**	local area network
ELAP	EtherTalk Link Access Protocol		**LAPB**	link access protocol-balanced
ELS	Event Logging System		**LAT**	local area transport
EST	Eastern Standard Time		**LCP**	Link Control Protocol
Eth	Ethernet		**LED**	light-emitting diode
fa-ga	functional address-group address		**LF**	largest frame; line feed
FCS	frame check sequence		**LLC**	logical link control
FECN	forward explicit congestion notification		**LLC2**	logical link control 2
FIFO	first in, first out		**LMI**	local management interface
FLT	filter library		**LRM**	LAN reporting mechanism
FR	Frame Relay		**LS**	link state
FRL	Frame Relay		**LSA**	link state advertisement
FTP	File Transfer Protocol		**LSB**	least significant bit
GMT	Greenwich Mean Time		**LSreq**	link state request
GOSIP	Government Open Systems Interconnection Profile		**LSrxl**	link state retransmission list
			LU	logical unit
GTE	General Telephone Company		**MAC**	medium access control
GWCON	Gateway Console		**Mb**	megabit
HDLC	high-level data link control		**MB**	megabyte
HEX	hexadecimal		**Mbps**	megabits per second
HPR	high-performance routing		**MBps**	megabytes per second
HST	TCP/IP host services		**MC**	multicast
HTF	host table format		**MCF**	MAC filtering
IBD	Integrated Boot Device		**MIB**	Management Information Base
ICMP	Internet Control Message Protocol		**MIB II**	Management Information Base II
ICP	Internet Control Protocol		**MILNET**	military network
ID	identification		**MOS**	Micro Operating System
IDP	Initial Domain Part		**MOSDDT**	Micro Operating System Dynamic Debugging Tool
IDP	Internet Datagram Protocol			
IEEE	Institute of Electrical and Electronics Engineers		**MOSPF**	Open Shortest Path First with multicast extensions
Ifc#	interface number		**MSB**	most significant bit

MSDU	MAC service data unit	**RISC**	reduced instruction-set computer
MTU	maximum transmission unit	**RNR**	receive not ready
nak	not acknowledged	**ROM**	read-only memory
NBP	Name Binding Protocol	**ROpcon**	Remote Operator Console
NBR	neighbor	**RPS**	ring parameter server
NCP	Network Control Protocol	**RTMP**	Routing Table Maintenance Protocol
NCP	Network Core Protocol	**RTP**	RouTing update Protocol
NetBIOS	Network Basic Input/Output System	**RTS**	request to send
NIST	National Institute of Standards and Technology	**Rtype**	route type
		rxmits	retransmissions
NPDU	Network Protocol Data Unit	**rxmt**	retransmit
NRZ	non-return-to-zero	**SAF**	source address filtering
NRZI	non-return-to-zero inverted	**SAP**	service access point
NSAP	Network Service Access Point	**SAP**	Service Advertising Protocol
NSF	National Science Foundation	**sdel**	start delimiter
NSFNET	National Science Foundation NETwork	**SDLC**	SDLC relay, synchronous data link control
NVCNFG	non-volatile configuration	**seqno**	sequence number
OPCON	Operator Console	**SGMP**	Simple Gateway Monitoring Protocol
OSI	open systems interconnection	**SL**	serial line
OSICP	OSI Control Protocol	**SMP**	standby monitor present
OSPF	Open Shortest Path First	**SMTP**	Simple Mail Transfer Protocol
OUI	organization unique identifier	**SNA**	Systems Network Architecture
PC	personal computer	**SNAP**	Subnetwork Access Protocol
PDN	public data network	**SNMP**	Simple Network Management Protocol
PING	Packet internet groper	**SNPA**	subnetwork point of attachment
PDU	protocol data unit	**SPF**	OSPF intra-area route
PID	process identification	**SPE1**	OSPF external route type 1
P-P	Point-to-Point	**SPE2**	OSPF external route type 2
PPP	Point-to-Point Protocol	**SPIA**	OSPF inter-area route type
PROM	programmable read-only memory	**SPID**	service profile ID
PU	physical unit	**SPX**	Sequenced Packet Exchange
PVC	permanent virtual circuit	**SQE**	signal quality error
RAM	random access memory	**SRAM**	static random access memory
RD	route descriptor	**SRB**	source routing bridge
REM	ring error monitor	**SRF**	specifically routed frame
REV	receive	**SRLY**	SDLC relay
RFC	Request for Comments	**SRT**	source routing transparent
RI	ring indicator; routing information	**SR-TB**	source routing-transparent bridge
RIF	routing information field	**STA**	static
RII	routing information indicator	**STB**	spanning tree bridge
RIP	Routing Information Protocol	**STE**	spanning tree explorer

STP	shielded twisted pair; spanning tree protocol	**UI**	unnumbered information
SVC	switched virtual circuit	**UTP**	unshielded twisted pair
TB	transparent bridge	**VINES**	VIrtual NEtworking System
TCN	topology change notification	**VIR**	variable information rate
TCP	Transmission Control Protocol	**VL**	virtual link
TCP/IP	Transmission Control Protocol/Internet Protocol	**VR**	virtual route
TEI	terminal point identifier	**WAN**	wide area network
TFTP	Trivial File Transfer Protocol	**WRS**	WAN restoral/reroute
TKR	token ring	**X.25**	packet-switched networks
TMO	timeout	**X.251**	X.25 physical layer
TOS	type of service	**X.252**	X.25 frame layer
TSF	transparent spanning frames	**X.253**	X.25 packet layer
TTL	time to live	**XID**	exchange identification
TTY	teletypewriter	**XNS**	Xerox Network Systems
TX	transmit	**XSUM**	checksum
UA	unnumbered acknowledgment	**ZIP**	AppleTalk Zone Information Protocol
UDP	User Datagram Protocol	**ZIP2**	AppleTalk Zone Information Protocol 2
		ZIT	Zone Information Table

Glossary

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.

- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

- Internet Request for Comments: 1208, *Glossary of Networking Terms*.

- Internet Request for Comments: 1392, *Internet Users' Glossary*.

- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

Contrast with: This refers to a term that has an opposed or substantively different meaning.

Synonym for: This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with: This is a backward reference from a defined term to all other terms that have the same meaning.

See: This refers the reader to multiple-word terms that have the same last word.

See also: This refers the reader to terms that have a related, but not synonymous, meaning.

A

AAL. ATM Adaptation Layer, the layer that adapts user data to/from the ATM network by adding/removing headers and segmenting/reassembling the data into/from cells.

AAL-5. ATM Adaptation Layer 5, one of several standard AALs. AAL-5 was designed for data communications, and is used by LAN Emulation and Classical IP.

abstract syntax. A data specification that includes all distinctions that are needed in data transmissions, but that omits (abstracts) other details such as those that depend on specific computer architectures. See also *abstract syntax notation 1 (ASN.1)* and *basic encoding rules (BER)*.

abstract syntax notation 1 (ASN.1). The Open Systems Interconnection (OSI) method for abstract syntax specified in the following standards:

- ITU-T Recommendation X.208 (1988) | ISO/IEC 8824: 1990

- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1: 1994

See also *basic encoding rules (BER)*.

ACCESS. In the Simple Network Management Protocol (SNMP), the clause in a Management Information Base (MIB) module that defines the minimum level of support that a managed node provides for an object.

acknowledgment. (1) The transmission, by a receiver, of acknowledge characters as an affirmative response to a sender. (T) (2) An indication that an item sent was received.

active monitor. In a token-ring network, a function performed at any one time by one ring station that initiates the transmission of tokens and provides token error recovery facilities. Any active adapter on the ring has the ability to provide the active monitor function if the current active monitor fails.

address. In data communication, the unique code assigned to each device, workstation, or user connected to a network.

address mapping table (AMT).   A table, maintained within the AppleTalk router, that provides a current mapping of node addresses to hardware addresses.

address mask.   For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address.   Synonymous with *subnet mask* and *subnetwork mask*.

address resolution.   (1) A method for mapping network-layer addresses to media-specific addresses. (2) See also *Address Resolution Protocol (ARP)* and *AppleTalk Address Resolution Protocol (AARP)*.

Address Resolution Protocol (ARP).   (1) In the Internet suite of protocols, the protocol that dynamically maps an IP address to an address used by a supporting metropolitan or local area network such as Ethernet or token-ring. (2) See also *Reverse Address Resolution Protocol (RARP)*.

addressing.   In data communication, the way in which a station selects the station to which it is to send data.

adjacent nodes.   Two nodes connected together by at least one path that connects no other node. (T)

Administrative Domain.   A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

Advanced Peer-to-Peer Networking (APPN).   An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; (c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

Advanced Peer-to-Peer Networking (APPN) end node.   A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services. An APPN end node can also attach to a subarea network as a peripheral node or to other end nodes.

Advanced Peer-to-Peer Networking (APPN) network.   A collection of interconnected network nodes and their client end nodes.

Advanced Peer-to-Peer Networking (APPN) network node.   A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server

- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service

- Session services for its local LUs and client end nodes

- Intermediate routing services within an APPN network

Advanced Peer-to-Peer Networking (APPN) node. An APPN network node or an APPN end node.

alert.   A message sent to a management services focal point in a network to identify a problem or an impending problem.

all-stations address.   In communications, synonym for *broadcast address*.

American National Standards Institute (ANSI).   An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

analog.   (1) Pertaining to data consisting of continuously variable physical quantities. (A) (2) Contrast with *digital*.

AppleTalk.   A network protocol developed by Apple Computer, Inc.   This protocol is used to interconnect network devices, which can be a mixture of Apple and non-Apple products.

AppleTalk Address Resolution Protocol (AARP).   In AppleTalk networks, a protocol that (a) translates AppleTalk node addresses into hardware addresses and (b) reconciles addressing discrepancies in networks that support more than one set of protocols.

AppleTalk Transaction Protocol (ATP).   In AppleTalk networks, a protocol that provides client/server request and response functions for hosts accessing the Zone Information Protocol (ZIP) for zone information.

APPN network.   See *Advanced Peer-to-Peer Networking (APPN) network*.

APPN network node.   See *Advanced Peer-to-Peer Networking (APPN) network node*.

arbitrary MAC addressing (AMA).   In DECnet architecture, an addressing scheme used by DECnet

Phase IV-Prime that supports universally administered addresses and locally administered addresses.

area. In Internet and DECnet routing protocols, a subset of a network or gateway grouped together by definition of the network administrator. Each area is self-contained; knowledge of an area's topology remains hidden from other areas.

asynchronous (ASYNC). Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T)

ATM. Asynchonous Transfer Mode, a connection-oriented, high-speed networking technology based on cell switching.

ATMARP. ARP in Classical IP.

attachment unit interface (AUI). In a local area network, the interface between the medium attachment unit and the data terminal equipment within a data station. (I)   (A)

authentication failure. In the Simple Network Management Protocol (SNMP), a trap that may be generated by an authentication entity when a requesting client is not a member of the SNMP community.

autonomous system. In TCP/IP, a group of networks and routers under one administrative authority. These networks and routers cooperate closely to propagate network reachability (and routing) information among themselves using an interior gateway protocol of their choice.

autonomous system number. In TCP/IP, a number assigned to an autonomous system by the same central authority that also assigns IP addresses. The autonomous system number makes it possible for automated routing algorithms to distinguish autonomous systems.

B

backbone. (1) In a local area network multiple-bridge ring configuration, a high-speed link to which the rings are connected by means of bridges or routers. A backbone may be configured as a bus or as a ring. (2) In a wide area network, a high-speed link to which nodes or data switching exchanges (DSEs) are connected.

backbone network. A central network to which smaller networks, normally of lower speed, connect. The backbone network usually has a much higher capacity than the networks it helps interconnect or is a wide-area network (WAN) such as a public packet-switched datagram network.

backbone router. (1) A router used to transmit data between areas. (2) One in a series of routers that is used to interconnect networks into a larger internet.

Bandwidth. The bandwidth of an optical link designates the information-carrying capacity of the link and is related to the maximum bit rate that a fiber link can support.

basic transmission unit (BTU). In SNA, the unit of data and control information passed between path control components. A BTU can consist of one or more path information units (PIUs).

baud. In asynchronous transmission, the unit of modulation rate corresponding to one unit interval per second; that is, if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud. (A)

bootstrap. (1) A sequence of instructions whose execution causes additional instructions to be loaded and executed until the complete computer program is in storage. (T) (2) A technique or device designed to bring itself into a desired state by means of its own action, for example, a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device. (A)

Border Gateway Protocol (BGP). An Internet Protocol (IP) routing protocol used between domains and autonomous systems.

border router. In Internet communications, a router, positioned at the edge of an autonomous system, that communicates with a router that is positioned at the edge of a different autonomous system.

bridge. A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address.

bridge identifier. An 8-byte field, used in a spanning tree protocol, composed of the MAC address of the port with the lowest port identifier and a user-defined value.

bridging. In LANs, the forwarding of a frame from one LAN segment to another. The destination is specified by the medium access control (MAC) sublayer address encoded in the destination address field of the frame header.

broadcast. (1) Transmission of the same data to all destinations. (T) (2) Simultaneous transmission of data to more than one destination. (3) Contrast with *multicast*.

broadcast address. In communications, a station address (eight 1's) reserved as an address common to

all stations on a link. Synonymous with *all-stations address*.

C

cache. (1) A special-purpose buffer storage, smaller and faster than main storage, used to hold a copy of instructions and data obtained from main storage and likely to be needed next by the processor. (T) (2) A buffer storage that contains frequently accessed instructions and data; it is used to reduce access time. (3) An optional part of the directory database in network nodes where frequently used directory information may be stored to speed directory searches. (4) To place, hide, or store in a cache.

call request packet. (1) A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network. (2) In X.25 communications, a call supervision packet transmitted by a DTE to ask for a call establishment through the network.

canonical address. In LANs, the IEEE 802.1 format for the transmission of medium access control (MAC) addresses for token-ring and Ethernet adapters. In canonical format, the least significant (rightmost) bit of each address byte is transmitted first. Contrast with *noncanonical address*.

carrier. An electric or electromagnetic wave or pulse train that may be varied by a signal bearing information to be transmitted over a communication system. (T)

carrier detect. Synonym for *received line signal detector (RLSD)*.

carrier sense. In a local area network, an ongoing activity of a data station to detect whether another station is transmitting. (T)

carrier sense multiple access with collision detection (CSMA/CD). A protocol that requires carrier sense and in which a transmitting data station that detects another signal while transmitting, stops sending, sends a jam signal, and then waits for a variable time before trying again. (T) (A)

channel. (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel service unit (CSU). A unit that provides the interface to a digital network. The CSU provides line conditioning (or equalization) functions, which keep the signal's performance consistent across the channel bandwidth; signal reshaping, which constitutes the binary pulse stream; and loopback testing, which

includes the transmission of test signals between the CSU and the network carrier's office channel unit. See also *data service unit (DSU)*.

checksum. (1) The sum of a group of data associated with the group and used for checking purposes. (T) (2) In error detection, a function of all bits in a block. If the written and calculated sums do not agree, an error is indicated. (3) On a diskette, data written in a sector for error detection purposes; a calculated checksum that does not match the checksum of data written in the sector indicates a bad sector. The data are either numeric or other character strings regarded as numeric for the purpose of calculating the checksum.

circuit switching. (1) A process that, on demand, connects two or more data terminal equipment (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (I) (A) (2) Synonymous with *line switching*.

class A network. In Internet communications, a network in which the high-order (most significant) bit of the IP address is set to 0 and the host ID occupies the three low-order octets.

class B network. In Internet communications, a network in which the two high-order (most significant and next-to-most significant) bits of the IP address are set to 1 and 0, respectively, and the host ID occupies the two low-order octets.

class of service (COS). A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The class of service is derived from a mode name specified by the initiator of a session.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user.

client/server. In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

clocking. (1) In binary synchronous communication, the use of clock pulses to control synchronization of data and control characters. (2) A method of controlling the number of data bits sent on a telecommunication line in a given time.

collision. An unwanted condition that results from concurrent transmissions on a channel. (T)

collision detection. In carrier sense multiple access with collision detection (CSMA/CD), a signal indicating that two or more stations are transmitting simultaneously.

Committed information rate. The maximum amount of data in bits that the network agrees to deliver.

community. In the Simple Network Management Protocol (SNMP), an administrative relationship between entities.

community name. In the Simple Network Management Protocol (SNMP), a string of octets identifying a community.

compression. (1) The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks. (2) Any encoding to reduce the number of bits used to represent a given message or record.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network.

configuration file. A file that specifies the characteristics of a system device or network.

configuration parameter. A variable in a configuration definition, the values of which can characterize the relationship of a product to other products in the same network or can define characteristics of the product itself.

configuration report server (CRS). In the IBM Token-Ring Network Bridge Program, the server that accepts commands from the LAN Network Manager (LNM) to get station information, set station parameters, and remove stations on its ring. This server also collects and forwards configuration reports generated by stations on its ring. The configuration reports include the new active monitor reports and the nearest active upstream neighbor (NAUN) reports.

congestion. See *network congestion*.

control point (CP). (1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network. (2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 subarea node, a network node control point (NNCP) in an APPN network node, and an end node control point (ENCP) in an APPN or LEN end node. An SSCP and an NNCP can provide services to other nodes.

control point management services (CPMS). A component of a control point, consisting of management services function sets, that provides facilities to assist in performing problem management, performance and accounting management, change management, and configuration management. Capabilities provided by the CPMS include sending requests to physical unit management services (PUMS) to test system resources, collecting statistical information (for example, error and performance data) from PUMS about the system resources, and analyzing and presenting test results and statistical information collected about the system resources. Analysis and presentation responsibilities for problem determination and performance monitoring can be distributed among multiple CPMSs.

control point management services unit (CP-MSU). The message unit that contains management services data and flows between management services function sets. This message unit is in general data stream (GDS) format. See also *management services unit (MSU)* and *network management vector transport (NMVT)*.

D

D-bit. Delivery-confirmation bit. In X.25 communications, the bit in a data packet or call-request packet that is set to 1 if end-to-end acknowledgment (delivery confirmation) is required from the recipient.

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

data carrier detect (DCD). Synonym for *received line signal detector (RLSD)*.

data circuit. (1) A pair of associated transmit and receive channels that provide a means of two-way data communication. (I) (2) In SNA, synonym for *link connection*. (3) See also *physical circuit* and *virtual circuit*.

Notes:

1. Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.

2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

data circuit-terminating equipment (DCE). In a data station, the equipment that provides the signal

conversion and coding between the data terminal equipment (DTE) and the line. (I)

Notes:

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.

2. A DCE may perform other functions that are usually performed at the network end of the line.

data link connection identifier (DLCI). The numeric identifier of a frame-relay subport or PVC segment in a frame-relay network. Each subport in a single frame-relay port has a unique DLCI. The following table, excerpted from the American National Standards Institute (ANSI) Standard T1.618 and the International Telegraph and Telephone Consultative Committee (ITU-T/CCITT) Standard Q.922, indicates the functions associated with certain DLCI values:

DLCI Values	Function
0	in-channel signaling
1–15	reserved
16–991	assigned using frame-relay connection procedures
992–1007	layer 2 management of frame-relay bearer service
1008–1022	reserved
1023	in-channel layer management

data link control (DLC). A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

Note: The DLC layer is usually independent of the physical transport mechanism and ensures the integrity of data that reaches the higher layers.

data link layer. In the Open Systems Interconnection reference model, the layer that provides services to transfer data between entities in the network layer over a communication link. The data link layer detects and possibly corrects errors that may occur in the physical layer. (T)

data link level. (1) In the hierarchical structure of a data station, the conceptual level of control or processing logic between high level logic and the data link that maintains control of the data link. The data link level performs such functions as inserting transmit bits and deleting receive bits; interpreting address and control fields; generating, transmitting, and interpreting commands and responses; and computing and interpreting frame check sequences. See also *packet level* and *physical level*. (2) In X.25 communications, synonym for *frame level*.

data link switching (DLSw). A method of transporting network protocols that use IEEE 802.2 logical link control (LLC) type 2. SNA and NetBIOS are examples of protocols that use LLC type 2. See also *encapsulation* and *spoofing*.

data packet. In X.25 communications, a packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface.

data service unit (DSU). A device that provides a digital data service interface directly to the data terminal equipment. The DSU provides loop equalization, remote and local testing capabilities, and a standard EIA/CCITT interface.

data set ready (DSR). Synonym for *DCE ready*.

data switching exchange (DSE). The equipment installed at a single location to provide switching functions, such as circuit switching, message switching, and packet switching. (I)

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both. (I) (A)

data terminal ready (DTR). A signal to the modem used with the EIA 232 protocol.

data transfer rate. The average number of bits, characters, or blocks per unit time passing between corresponding equipment in a data transmission system. (I)

Notes:

1. The rate is expressed in bits, characters, or blocks per second, minute, or hour.

2. Corresponding equipment should be indicated; for example, modems, intermediate equipment, or source and sink.

datagram. (1) In packet switching, a self-contained packet, independent of other packets, that carries information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the DTEs and the network. (I) (2) In TCP/IP, the basic unit of information passed across the Internet environment. A datagram contains a source and destination address along with the data. An Internet Protocol (IP) datagram consists of an IP header followed by the transport layer data. (3) See also *packet* and *segment*.

Datagram Delivery Protocol (DDP). In AppleTalk networks, a protocol that provides network connectivity

by means of connectionless socket-to-socket delivery service on the internet layer.

DCE ready. In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that the local data circuit-terminating equipment (DCE) is connected to the communication channel and is ready to send data. Synonymous with *data set ready (DSR)*.

DECnet. A network architecture that defines the operation of a family of software modules, databases, and hardware components typically used to tie Digital Equipment Corporation systems together for resource sharing, distributed computation, or remote system configuration. DECnet network implementations follow the Digital Network Architecture (DNA) model.

default. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

designated router. A router that informs end nodes of the existence and identity of other routers. The selection of the designated router is based upon the router with the highest priority. When several routers share the highest priority, the router with the highest station address is selected.

destination node. The node to which a request or data is sent.

destination port. The 8-port asynchronous adapter that serves as a connection point with a serial service.

destination service access point (DSAP). In SNA and TCP/IP, a logical address that allows a system to route data from a remote device to the appropriate communications support. Contrast with *source service access point (SSAP)*.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

digital. (1) Pertaining to data that consist of digits. (T) (2) Pertaining to data in the form of digits. (A) (3) Contrast with *analog*.

Digital Network Architecture (DNA). The model for all DECnet hardware and software implementations.

direct memory access (DMA). The system facility that allows a device on the Micro Channel bus to get direct access to the system or bus memory without the intervention of the system processor.

directory. A table of identifiers and references to the corresponding items of data. (I) (A)

directory service (DS). An application service element that translates the symbolic names used by application

processes into the complete network addresses used in an OSI environment. (T)

directory services (DS). A control point component of an APPN node that maintains knowledge of the location of network resources.

disable. To make nonfunctional.

disabled. (1) Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line.

domain. (1) That part of a computer network in which the data processing resources are under common control. (T) (2) In Open Systems Interconnection (OSI), a part of a distributed system or a set of managed objects to which a common policy applies. (3) See *Administrative Domain* and *domain name*.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is `ralvm7.vnet.ibm.com`, each of the following is a domain name:

- `ralvm7.vnet.ibm.com`
- `vnet.ibm.com`
- `ibm.com`

domain name server. In the Internet suite of protocols, a server program that supplies name-to-address translation by mapping domain names to IP addresses. Synonymous with *name server*.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

dump. (1) Data that has been dumped. (T) (2) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

dynamic reconfiguration (DR). The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables or deactivating the affected major node.

Dynamic Routing. Routing using learned routes rather than routes statically configured at initialization.

E

echo. In data communication, a reflected signal on a communications channel. For example, on a communications terminal, each signal is displayed twice, once when entered at the local terminal and again when returned over the communications link. This allows the signals to be checked for accuracy.

EIA 232. In data communication, a specification of the Electronic Industries Association (EIA) that defines the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE), using serial binary data interchange.

Electronic Industries Association (EIA). An organization of electronics manufacturers that advances the technological growth of the industry, represents the views of its members, and develops industry standards.

encapsulation. (1) In communications, a technique used by layered protocols by which a layer adds control information to the protocol data unit (PDU) from the layer it supports. In this respect, the layer encapsulates the data from the supported layer. In the Internet suite of protocols, for example, a packet would contain control information from the physical layer, followed by control information from the network layer, followed by the application protocol data. (2) See also *data link switching*.

encode. To convert data by the use of a code in such a manner that reconversion to the original form is possible. (T)

end node (EN). (1) See *Advanced Peer-to-Peer Networking (APPN) end node* and *low-entry networking (LEN) end node*. (2) In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

entry point (EP). In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

Ethernet. A 10-Mbps baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and delayed retransmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file.

exception response (ER). In SNA, a protocol requested in the form-of-response-requested field of a request header that directs the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, a negative response, but not a positive response, can be returned. Contrast with *definite response* and *no response*.

exchange identification (XID). A specific type of basic link unit that is used to convey node and link characteristics between adjacent nodes. XIDs are exchanged between link stations before and during link activation to establish and negotiate link and node characteristics, and after link activation to communicate changes in these characteristics.

explicit route (ER). In SNA, a series of one or more transmission groups that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*.

explorer frame. See *explorer packet*.

explorer packet. In LANs, a packet that is generated by the source host and that traverses the entire source routing part of a LAN, gathering information on the possible paths available to the host.

exterior gateway. In Internet communications, a gateway on one autonomous system that communicates with another autonomous system. Contrast with *interior gateway*.

Exterior Gateway Protocol (EGP). In the Internet suite of protocols, a protocol, used between domains and autonomous systems, that enables network reachability information to be advertised and exchanged. IP network addresses in one autonomous system are advertised to another autonomous system by means of EGP-participating routers. An example of an EGP is the Border Gateway Protocol (BGP). Contrast with Interior Gateway Protocol (IGP).

F

File Transfer Protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

flow control. (1) In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. (2) See also *pacing*.

fragment. See *fragmentation*.

fragmentation. (1) The process of dividing a datagram into smaller parts, or fragments, to match the capabilities of the physical medium over which it is to be transmitted. (2) See also *segmenting*.

frame. (1) In Open Systems Interconnection architecture, a data structure pertaining to a particular area of knowledge and consisting of slots that can accept the values of specific attributes and from which inferences can be drawn by appropriate procedural attachments. (T) (2) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (3) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

frame level. Synonymous with *data link level*. See *link level*.

frame relay. (1) An interface standard describing the boundary between a user's equipment and a fast-packet network. In frame-relay systems, flawed frames are discarded; recovery comes end-to-end rather than hop-by-hop. (2) A technique derived from the integrated services digital network (ISDN) D channel standard. It assumes that connections are reliable and dispenses with the overhead of error detection and control within the network.

G

gateway. (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (2) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols. (3) In TCP/IP, synonym for *router*.

general data stream (GDS). The data stream used for conversations in LU 6.2 sessions.

general data stream (GDS) variable. A type of RU substructure that is preceded by an identifier and a length field and includes either application data, user control data, or SNA-defined control data.

H

header. (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

heap memory. The amount of RAM used to dynamically allocate data structures.

Hello. A protocol used by a group of cooperating, trusting routers to allow them to discover minimal delay routes.

hello message. (1) A message sent periodically to establish and test reachability between routers or between routers and hosts. (2) In the Internet suite of protocols, a message defined by the Hello protocol as an Interior Gateway Protocol (IGP).

heuristic. Pertaining to exploratory methods of problem solving in which solutions are discovered by evaluation of the progress made toward the final result.

high-level data link control (HDLC). In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

high-performance routing (HPR). An addition to the Advanced Peer-to-Peer Networking (APPN) architecture that enhances data routing performance and reliability, especially when using high-speed links.

hop. (1) In APPN, a portion of a route that has no intermediate nodes. It consists of only a single transmission group connecting adjacent nodes. (2) To the routing layer, the logical distance between two nodes in a network.

hop count. (1) A metric or measure of distance between two points. (2) In Internet communications, the number of routers that a datagram passes through on its way to its destination. (3) In SNA, a measure of the number of links to be traversed in a path to a destination.

host. In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe.

hysteresis. The amount the temperature must change past the set alert threshold before the alert condition is cleared.

I

I-frame. Information frame.

information (I) frame. A frame in I format used for numbered information transfer.

input/output channel. In a data processing system, a functional unit that handles transfer of data between internal and peripheral equipment. (I) (A)

integrated services digital network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

Note: ISDNs are used in public and private network architectures.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T) (2) Hardware, software, or both, that links systems, programs, or devices.

interior gateway. In Internet communications, a gateway that communicates only with its own autonomous system. Contrast with *exterior gateway*.

Interior Gateway Protocol (IGP). In the Internet suite of protocols, a protocol used to propagate network reachability and routing information within an autonomous system. Examples of IGPs are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

intermediate node. A node that is at the end of more than one branch. (T)

intermediate session routing (ISR). A type of routing function within an APPN network node that provides session-level flow control and outage reporting for all sessions that pass through the node but whose end points are elsewhere.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

International Telecommunication Union (ITU). The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address. See *IP address*.

Internet Architecture Board (IAB). The technical body that oversees the development of the Internet suite of protocols that are known as TCP/IP.

Internet Control Message Protocol (ICMP). The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source. ICMP is part of the Internet Protocol.

Internet Control Protocol (ICP). The VIrtual NEtworking System (VINES) protocol that provides exception notifications, metric notifications, and PING support. See also *RouTing update Protocol (RTP)*.

Internet Engineering Task Force (IETF). The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet.

Internet Protocol (IP). A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internetwork Packet Exchange (IPX). (1) The network protocol used to connect Novell's servers, or any workstation or router that implements IPX, with other workstations. Although similar to the Internet Protocol (IP), IPX uses different packet formats and terminology. (2) See also *Xerox Network Systems (XNS)*.

interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units. (T)

intra-area routing. In Internet communications, the routing of data within an area.

Inverse Address Resolution Protocol (InARP). In the Internet suite of protocols, the protocol used for locating a protocol address through the known hardware

address. In a frame-relay context, the data link connection identifier (DLCI) is synonymous with the known hardware address.

IP address. The 32-bit address defined by the Internet Protocol, standard 5, Request for Comments (RFC) 791. It is usually represented in dotted decimal notation.

IP datagram. In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

IP router. A device in an IP internet that is responsible for making decisions about the paths over which network traffic will flow. Routing protocols are used to gain information about the network and to determine the best route over which the datagram should be forwarded toward the final destination. The datagrams are routed based on IP destination addresses.

IPXWAN. A Novell protocol that is used to exchange router-to-router information before exchanging standard Internetwork Packet Exchange (IPX) routing information and traffic over wide area networks (WANs).

L

LAN bridge server (LBS). In the IBM Token-Ring Network Bridge Program, the server that keeps statistical information about frames forwarded between two or more rings (through a bridge). The LBS sends these statistics to the appropriate LAN managers through the LAN reporting mechanism (LRM).

LAN Emulation (LE). An ATM Forum standard that supports legacy LAN applications over ATM networks.

LAN Emulation Client (LEC). A LAN Emulation component that represents users of the Emulated LAN.

LAN Emulation Configuration Server (LECS). A LAN Emulation Service component that centralizes and disseminates configuration data.

LAN Emulation Server (LES). A LAN Emulation Service component that resolves LAN Destinations to ATM Addresses.

LAN Network Manager (LNM). An IBM licensed program that enables a user to manage and monitor LAN resources from a central workstation.

LAN segment. (1) Any portion of a LAN (for example, a bus or ring) that can operate independently, but that

is connected to other parts of the network by means of bridges. (2) A ring or bus network without bridges.

layer. (1) In network architecture, a group of services that is complete from a conceptual point of view, that is one out of a set of hierarchically arranged groups, and that extends across all systems that conform to the network architecture. (T) (2) In the Open Systems Interconnection reference model, one of seven conceptually complete, hierarchically arranged groups of services, functions, and protocols, that extend across all open systems. (T) (3) In SNA, a grouping of related functions that are logically separate from the functions in other groups. Implementation of the functions in one layer can be changed without affecting functions in other layers.

LE. LAN Emulation. An ATM Forum standard that supports legacy LAN applications over ATM networks.

LEC. LAN Emulation Client. A LAN Emulation component that represents users of the Emulated LAN.

LECS. LAN Emulation Configuration Server. A LAN Emulation Service component that centralizes and disseminates configuration data.

LES. LAN Emulation Server. A LAN Emulation Service component that resolves LAN Destinations to ATM Addresses.

line switching. Synonym for *circuit switching*.

link. The combination of the link connection (the transmission medium) and two link stations, one at each end of the link connection. A link connection can be shared among multiple links in a multipoint or token-ring configuration.

link access protocol balanced (LAPB). A protocol used for accessing an X.25 network at the link level. LAPB is a duplex, asynchronous, symmetric protocol, used in point-to-point communication.

link-attached. (1) Pertaining to devices that are connected to a controlling unit by a data link. (2) Contrast with *channel-attached*. (3) Synonymous with *remote*.

link connection. (1) The physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit-terminating equipment (DCE). (2) In SNA, synonymous with *data circuit*.

link level. (1) A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the full-duplex link connecting the subscriber's machine to the network node. LAP and

LAPB are the link access protocols recommended by the CCITT. (2) See *data link level*.

link-state. In routing protocols, the advertised information about the usable interfaces and reachable neighbors of a router or network. The protocol's topological database is formed from the collected link-state advertisements.

link station. (1) The hardware and software components within a node representing a connection to an adjacent node over a specific link. For example, if node A is the primary end of a multipoint line that connects to three adjacent nodes, node A will have three link stations representing the connections to the adjacent nodes. (2) See also *adjacent link station (ALS)*.

local. (1) Pertaining to a device accessed directly without use of a telecommunication line. (2) Contrast with *remote*. (3) Synonym for *channel-attached*.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. (3) See also *Ethernet* and *token ring*. (4) Contrast with *metropolitan area network (MAN)* and *wide area network (WAN)*.

local bridging. A function of a bridge program that allows a single bridge to connect multiple LAN segments without using a telecommunication link. Contrast with *remote bridging*.

local management interface (LMI). See *local management interface (LMI) protocol*.

local management interface (LMI) protocol. In NCP, a set of frame-relay network management procedures and messages used by adjacent frame-relay nodes to exchange line status information over DLCI X'00'. NCP supports both the American National Standards Institute (ANSI) and International Telegraph and Telephone Consultative Committee (ITU-T/CCITT) versions of LMI protocol. These standards refer to LMI protocol as *link integrity verification tests (LIVT)*.

locally administered address. In a local area network, an adapter address that the user can assign to override the universally administered address. Contrast with *universally administered address*.

logical channel. In packet mode operation, a sending channel and a receiving channel that together are used to send and receive data over a data link at the same time. Several logical channels can be established on the same data link by interleaving the transmission of packets.

logical link. A pair of link stations, one in each of two adjacent nodes, and their underlying link connection, providing a single link-layer connection between the two nodes. Multiple logical links can be distinguished while they share the use of the same physical media connecting two nodes. Examples are 802.2 logical links used on local area network (LAN) facilities and LAP E logical links on the same point-to-point physical link between two nodes. The term logical link also includes the multiple X.25 logical channels that share the use of the access link from a DTE to an X.25 network.

logical link control (LLC). The data link control (DLC) LAN sublayer that provides two types of DLC operation for the orderly exchange of information. The first type is connectionless service, which allows information to be sent and received without establishing a link. The LLC sublayer does not perform error recovery or flow control for connectionless service. The second type is connection-oriented service, which requires establishing a link prior to the exchange of information. Connection-oriented service provides sequenced information transfer, flow control, and error recovery.

logical link control (LLC) protocol. In a local area network, the protocol that governs the exchange of transmission frames between data stations independently of how the transmission medium is shared. (T) The LLC protocol was developed by the IEEE 802 committee and is common to all LAN standards.

logical link control (LLC) protocol data unit. A unit of information exchanged between link stations in different nodes. The LLC protocol data unit contains a destination service access point (DSAP), a source service access point (SSAP), a control field, and user data.

logical unit (LU). A type of network accessible unit that enables users to gain access to network resources and communicate with each other.

loopback test. A test in which signals from a tester are looped at a modem or other network element back to the tester for measurements that determine or verify the quality of the communications path.

low-entry networking (LEN). A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between logical units.

low-entry networking (LEN) end node. A LEN node receiving network services from an adjacent APPN network node.

low-entry networking (LEN) node. A node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

M

Management Information Base (MIB). (1) A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed. (3) In OSI, the conceptual repository of management information within an open system.

management station. In Internet communications, the system responsible for managing all, or a portion of, a network. The management station communicates with network management agents that reside in the managed node by means of a network management protocol, such as the Simple Network Management Protocol (SNMP).

mapping. The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

mask. (1) A pattern of characters used to control retention or elimination of portions of another pattern of characters. (I) (A) (2) To use a pattern of characters to control retention or elimination of portions of another pattern of characters. (I) (A)

maximum transmission unit (MTU). In LANs, the largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the MTU for Ethernet is 1500 bytes.

medium access control (MAC). In LANs, the sublayer of the data link control layer that supports medium-dependent functions and uses the services of the physical layer to provide services to the logical link control (LLC) sublayer. The MAC sublayer includes the method of determining when a device has access to the transmission medium.

medium access control (MAC) protocol. In a local area network, the protocol that governs access to the transmission medium, taking into account the topological aspects of the network, in order to enable the exchange of data between data stations. (T)

medium access control (MAC) sublayer. In a local area network, the part of the data link layer that applies a medium access method. The MAC sublayer supports topology-dependent functions and uses the services of the physical layer to provide services to the logical link control sublayer. (T)

metric. In Internet communications, a value, associated with a route, which is used to discriminate between multiple exit or entry points to the same autonomous system. The route with the lowest metric is preferred.

metropolitan area network (MAN). A network formed by the interconnection of two or more networks which may operate at higher speed than those networks, may cross administrative boundaries, and may use multiple access methods. (T) Contrast with *local area network (LAN)* and *wide area network (WAN)*.

MIB object. Synonym for *MIB variable*.

MIB variable. In the Simple Network Management Protocol (SNMP), a specific instance of data defined in a MIB module. Synonymous with *MIB object*.

MIB view. In the Simple Network Management Protocol (SNMP), the collection of managed objects, known to the agent, that is visible to a particular community.

MILNET. The military network that was originally part of ARPANET. It was partitioned from ARPANET in 1984. MILNET provides a reliable network service for military installations.

modem (modulator/demodulator). (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

modulo. (1) Pertaining to a modulus; for example, 9 is equivalent to 4 modulo 5. (2) See also *modulus*.

modulus. A number, such as a positive integer, in a relationship that divides the difference between two related numbers without leaving a remainder; for example, 9 and 4 have a modulus of 5 (9 - 4 = 5; 4 - 9 = -5; and 5 divides both 5 and -5 without leaving a remainder).

monitor. (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T) (2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A) (3) The function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

multicast. (1) Transmission of the same data to a selected group of destinations. (T) (2) A special form of broadcast in which copies of a packet are delivered to only a subset of all possible destinations.

multiple-domain support (MDS). A technique for transporting management services data between management services function sets over LU-LU and CP-CP sessions. See also *multiple-domain support message unit (MDS-MU)*.

multiple-domain support message unit (MDS-MU). The message unit that contains management services data and flows between management services function sets over the LU-LU and CP-CP sessions used by multiple-domain support. This message unit, as well as the actual management services data that it contains, is in general data stream (GDS) format. See also *control point management services unit (CP-MSU)*, *management services unit (MSU)*, and *network management vector transport (NMVT)*.

N

Name Binding Protocol (NBP). In AppleTalk networks, a protocol that provides name translation function from the AppleTalk entity (resource) name (character string) into an AppleTalk IP address (16-bit number) on the transport layer.

name resolution. In Internet communications, the process of mapping a machine name to the corresponding Internet Protocol (IP) address. See also *Domain Name System (DNS)*.

name server. In the Internet suite of protocols, synonym for *domain name server*.

nearest active upstream neighbor (NAUN). In the IBM Token-Ring Network, the station sending data directly to a given station on the ring.

neighbor. A router on a common subnetwork that has been designated by a network administrator to receive routing information.

NetBIOS. Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols.

network. (1) A configuration of data processing devices and software connected for information interchange. (2) A group of nodes and the links interconnecting them.

network accessible unit (NAU). A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with *network addressable unit*.

network address. According to ISO 7498-3, a name, unambiguous within the OSI environment, that identifies a set of network service access points.

network addressable unit (NAU). Synonym for *network accessible unit*.

network architecture. The logical structure and operating principles of a computer network. (T)

Note: The operating principles of a network include those of services, functions, and protocols.

network congestion. An undesirable overload condition caused by traffic in excess of what a network can handle.

network identifier. (1) In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C). (2) A 1- to 8-byte customer-selected name or an 8-byte IBM-registered name that uniquely identifies a specific subnetwork.

Network Information Center (NIC). In Internet communications, local, regional, and national groups throughout the world who provide assistance, documentation, training, and other services to users.

network layer. In Open Systems Interconnection (OSI) architecture, the layer that is responsible for routing, switching, and link-layer access across the OSI environment.

network management. The process of planning, organizing, and controlling a communication-oriented data processing or information system.

network management station. In the Simple Network Management Protocol (SNMP), a station that executes management application programs that monitor and control network elements.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network manager. A program or group of programs that is used to monitor, manage, and diagnose the problems of a network.

network node (NN). See *Advanced Peer-to-Peer Networking (APPN) network node*.

network user address (NUA). In X.25 communications, the X.121 address containing up to 15 binary code digits.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) Any device, attached to a network, that transmits and receives data.

noncanonical address. In LANs, a format for the transmission of medium access control (MAC) addresses for token-ring adapters. In noncanonical format, the most significant (leftmost) bit of each address byte is transmitted first. Contrast with *canonical address*.

nonseed router. In AppleTalk networks, a router that acquires network number range and zone list information from a seed router attached to the same network.

O

Open Shortest Path First (OSPF). In the Internet suite of protocols, a function that provides intradomain information transfer. An alternative to the Routing Information Protocol (RIP), OSPF allows the lowest-cost routing and handles routing in large regional or corporate networks.

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information. (T) (A) (2) The use of standardized procedures to enable the interconnection of data processing systems.

Note: OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

Open Systems Interconnection (OSI) architecture. Network architecture that adheres to that particular set of ISO standards that relates to Open Systems Interconnection. (T)

Open Systems Interconnection (OSI) reference model. A model that describes the general principles of the Open Systems Interconnection, as well as the purpose and the hierarchical arrangement of its seven layers. (T)

origin. An external logical unit (LU) or application program from which a message or other data originates. See also *destination*.

orphan circuit. A non-configured circuit whose availability is learned dynamically.

P

pacing. (1) A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. (2) See also *flow control*, *receive pacing*, *send pacing*, *session-level pacing*, and *virtual route (VR) pacing*.

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

packet internet groper (PING). (1) In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations by sending the destinations an Internet Control Message Protocol (ICMP) echo request and waiting for a reply. (2) In communications, a test of reachability.

packet mode operation. Synonym for *packet switching*.

packet switching. (1) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (I) (2) Synonymous with *packet mode operation*. See also *circuit switching*.

parallel bridges. A pair of bridges connected to the same LAN segment, creating redundant paths to the segment.

parallel transmission groups. Multiple transmission groups between adjacent nodes, with each group having a distinct transmission group number.

path. (1) In a network, any route between any two nodes. A path may include more than one branch. (T) (2) The series of transport network components (path control and data link control) that are traversed by the information exchanged between two network accessible units. See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*.

path control (PC). The function that routes message units between network accessible units in the network and provides the paths between them. It converts the basic information units (BIUs) from transmission control (possibly segmenting them) into path information units

(PIUs) and exchanges basic transmission units containing one or more PIUs with data link control. Path control differs by node type: some nodes (APPN nodes, for example) use locally generated session identifiers for routing, and others (subarea nodes) use network addresses for routing.

path cost. In link-state routing protocols, the sum of the link costs along the path between two nodes or networks.

path information unit (PIU). A message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment.

pattern-matching character. A special character such as an asterisk (*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace a pattern-matching character. Synonymous with *global character* and *wildcard character*.

permanent virtual circuit (PVC). In X.25 and frame-relay communications, a virtual circuit that has a logical channel permanently assigned to it at each data terminal equipment (DTE). Call-establishment protocols are not required. Contrast with *switched virtual circuit (SVC)*.

physical circuit. A circuit established without multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

physical layer. In the Open Systems Interconnection reference model, the layer that provides the mechanical, electrical, functional, and procedural means to establish, maintain, and release physical connections over the transmission medium. (T)

physical unit (PU). (1) The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. (2) See also *peripheral PU* and *subarea PU*.

ping command. The command that sends an Internet Control Message Protocol (ICMP) echo-request packet to a gateway, router, or host with the expectation of receiving a reply.

Point-to-Point Protocol (PPP). A protocol that provides a method for encapsulating and transmitting packets over serial point-to-point links.

polling. (1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (I) (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A)

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine. (6) Synonymous with *socket*.

port number. In Internet communications, the identification of an application entity to the transport service.

problem determination. The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environmental failure such as a power loss, or user error.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

protocol. (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In Open Systems Interconnection architecture, a set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T) (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See *bracket protocol* and *link protocol*.

protocol data unit (PDU). A unit of data specified in a protocol of a given layer and consisting of protocol control information of this layer, and possibly user data of this layer. (T)

R

Rapid Transport Protocol (RTP) connection. In high-performance routing (HPR), the connection established between the endpoints of the route to transport session traffic.

reachability. The ability of a node or a resource to communicate with another node or resource.

read-only memory (ROM). Memory in which stored data cannot be modified by the user except under special conditions.

reassembly. In communications, the process of putting segmented packets back together after they have been received.

receive not ready (RNR). In communications, a data link command or response that indicates a temporary condition of being unable to accept incoming frames.

receive not ready (RNR) packet. See *RNR packet*.

received line signal detector (RLSD). In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that it is receiving a signal from the remote data circuit-terminating equipment (DCE). Synonymous with *carrier detect* and *data carrier detect (DCD)*.

Recognized Private Operating Agency (RPOA). Any individual, company, or corporation, other than a government department or service, that operates a telecommunication service and is subject to the obligations undertaken in the Convention of the International Telecommunication Union and in the Regulations; for example, a communication common carrier.

reduced instruction-set computer (RISC). A computer that uses a small, simplified set of frequently used instructions for rapid execution.

remote. (1) Pertaining to a system, program, or device that is accessed through a telecommunication line. (2) Synonym for *link-attached*. (3) Contrast with *local*.

remote bridging. The function of a bridge that allows two bridges to connect multiple LANs using a telecommunication link. Contrast with *local bridging*.

Remote Execution Protocol (REXEC). A protocol that allows the execution of a command or program on any host in the network. The local host receives the results of the command execution.

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

reset. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

reset request packet. In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to the data circuit-terminating equipment (DCE) to request that a virtual call or a permanent virtual circuit be reset. The reason for the request can also be specified in the packet.

ring. See *ring network*.

ring network. (1) A network in which every node has exactly two branches connected to it and in which there are exactly two paths between any two nodes. (T) (2) A network configuration in which devices are connected by unidirectional transmission links to form a closed path.

ring segment. A section of a ring that can be isolated (by unplugging connectors) from the rest of the ring. See *LAN segment*.

rlogin (remote login). A service, offered by Berkeley UNIX-based systems, that allows authorized users of one machine to connect to other UNIX systems across an internet and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

RNR packet. A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate a temporary inability to accept additional packets for a virtual call or permanent virtual circuit.

root bridge. The bridge that is the root of a spanning tree formed between other active bridges in the bridging network. The root bridge originates and transmits bridge protocol data units (BPDUs) to other active bridges to maintain the spanning tree topology. It is the bridge with the highest priority in the network.

route. (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

route bridge. A function of an IBM bridge program that allows two bridge computers to use a telecommunication link to connect two LANs. Each bridge computer is connected directly to one of the LANs, and the telecommunication link connects the two bridge computers.

route extension (REX).   In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node.   See also *explicit route (ER)*, *path*, and *virtual route (VR)*.

Route Selection control vector (RSCV).   A control vector that describes a route within an APPN network. The RSCV consists of an ordered sequence of control vectors that identify the TGs and nodes that make up the path from an origin node to a destination node.

router.   (1) A computer that determines the path of network traffic flow.   The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses.   (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer.   (3) In OSI terminology, a function that determines a path by which an entity can be reached.   (4) In TCP/IP, synonymous with *gateway*. (5) Contrast with *bridge*.

routing.   (1) The assignment of the path by which a message is to reach its destination.   (2) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

routing domain.   In Internet communications, a group of intermediate systems that use a routing protocol so that the representation of the overall network is the same within each intermediate system.   Routing domains are connected to each other by exterior links.

Routing Information Protocol (RIP).   In the Internet suite of protocols, an interior gateway protocol used to exchange intradomain routing information and to determine optimum routes between internet hosts.   RIP determines optimum routes on the basis of route metrics, not link transmission speed.

routing loop.   A situation that occurs when routers circulate information among themselves until convergence occurs or until the networks involved are considered unreachable.

routing protocol.   A technique used by a router to find other routers and to remain up to date about the best way to get to reachable networks.

routing table.   A collection of routes used to direct datagram forwarding or to establish a connection.   The information is passed among routers to identify network topology and destination feasibility.

Routing Table Maintenance Protocol (RTMP).   In AppleTalk networks, a protocol that provides routing information generation and maintenance on the transport layer by means of the AppleTalk routing table. The AppleTalk routing table directs packet transmission through the internet from source socket to destination socket.

RouTing update Protocol (RTP).   The VIrtual NEtworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes.   See also *Internet Control Protocol (ICP)*.

rsh.   A variant of the rlogin command that invokes a command interpreter on a remote UNIX machine and passes the command-line arguments to the command interpreter, skipping the login step completely.

S

seed router.   In AppleTalk networks, a router that maintains configuration data (network range numbers and zone lists, for example) for the network.   Each network must have at least one seed router.   The seed router must be initially set up using the configurator tool. Contrast with *nonseed router*.

segment.   (1) A section of cable between components or devices.   A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected.   (2) In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data.

segmenting.   In OSI, a function performed by a layer to map one protocol data unit (PDU) from the layer it supports into multiple PDUs.

sequence number.   In communications, a number assigned to a particular frame or packet to control the transmission flow and receipt of data.

server.   A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server.   (T)

service access point (SAP).   (1) In Open Systems Interconnection (OSI) architecture, the point at which the services of a layer are provided by an entity of that layer to an entity of the next higher layer.   (T)   (2) A logical point made available by an adapter where information can be received and transmitted.   A single service access point can have many links terminating in it.

Service Advertising Protocol (SAP). In Internetwork Packet Exchange (IPX), a protocol that provides the following:

- A mechanism that allows IPX servers on an internet to advertise their services by name and type. Servers using this protocol have their name, service type, and address recorded in all file servers running NetWare.
- A mechanism that allows a workstation to broadcast a query to discover the identities of all servers of all types, all servers of a specific type, or the nearest server of a specific type.
- A mechanism that allows a workstation to query any file server running NetWare to discover the names and addresses of all servers of a specific type.

session. (1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection. (T) (2) A logical connection between two network accessible units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) accompanying any transmissions exchanged during the session.

Simple Network Management Protocol (SNMP). In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

SNA management services (SNA/MS). The services provided to assist in management of SNA networks.

socket. An endpoint for communication between processes or application programs.

source route bridging. In LANs, a bridging method that uses the routing information field in the IEEE 802.5 medium access control (MAC) header of a frame to determine which rings or token-ring segments the frame must transit. The routing information field is inserted into the MAC header by the source node. The information in the routing information field is derived from explorer packets generated by the source host.

source routing. In LANs, a method by which the sending station determines the route the frame will follow and includes the routing information with the frame. Bridges then read the routing information to determine whether they should forward the frame.

source service access point (SSAP). In SNA and TCP/IP, a logical address that allows a system to send data to a remote device from the appropriate communications support. Contrast with *destination service access point (DSAP)*.

spanning tree. In LAN contexts, the method by which bridges automatically develop a routing table and update that table in response to changing topology to ensure that there is only one route between any two LANs in the bridged network. This method prevents packet looping, where a packet returns in a circuitous route back to the sending router.

sphere of control (SOC). The set of control point domains served by a single management services focal point.

sphere of control (SOC) node. A node directly in the sphere of control of a focal point. A SOC node has exchanged management services capabilities with its focal point. An APPN end node can be a SOC node if it supports the function to exchange management services capabilities.

split horizon. A technique for minimizing the time to achieve network convergence. A router records the interface over which it received a particular route and does not propagate its information about the route back over the same interface.

spoofing. For data links, a technique in which a protocol initiated from an end station is acknowledged and processed by an intermediate node on behalf of the final destination. In IBM 6611 data link switching, for example, SNA frames are encapsulated into TCP/IP packets for transport across a non-SNA wide area network, unpacked by another IBM 6611, and passed to the final destination. A benefit of spoofing is the prevention of end-to-end session timeouts.

standard MIB. In the Simple Network Management Protocol (SNMP), a MIB module that is located under the management branch of the Structure of Management Information (SMI) and that is considered a standard by the Internet Engineering Task Force (IETF).

static route. The route between hosts, networks, or both that is manually entered into a routing table.

station. An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

StreetTalk. In the VIrtual NEtworking System (VINES), a unique network-wide naming and addressing system that allows users to locate and access any resource on the network without knowing the network topology. See also *Internet Control Protocol (ICP)* and *RouTing update Protocol (RTP)*.

Structure of Management Information (SMI). (1) In the Simple Network Management Protocol (SNMP), the rules used to define the objects that can be accessed by means of a network management protocol. (2) In OSI, the set of standards relating to management information. The set includes the *Management Information Model* and the *Guidelines for the Definition of Managed Objects*.

subarea. A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subnet. (1) In TCP/IP, a part of a network that is identified by a portion of the IP address. (2) Synonym for *subnetwork*.

subnet address. In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

subnet mask. Synonym for *address mask*.

subnetwork. (1) Any group of nodes that have a set of common characteristics, such as the same network ID. (2) Synonymous with *subnet*.

Subnetwork Access Protocol (SNAP). In LANs, a 5-byte protocol discriminator that identifies the non-IEEE standard protocol family to which a packet belongs. The SNAP value is used to differentiate between protocols that use $AA as their service access point (SAP) value.

subnetwork mask. Synonym for *address mask*.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

switched virtual circuit (SVC). An X.25 circuit that is dynamically established when needed. The X.25 equivalent of a switched line. Contrast with *permanent virtual circuit (PVC)*.

synchronous. (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). (1) A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I) (2) Contrast with *binary synchronous communication (BSC)*.

SYNTAX. In the Simple Network Management Protocol (SNMP), a clause in the MIB module that defines the abstract data structure that corresponds to a managed object.

system configuration. A process that specifies the devices and programs that form a particular data processing system.

system services control point (SSCP). A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

T

Telnet. In the Internet suite of protocols, a protocol that provides remote terminal connection service. It allows users of one host to log on to a remote host and interact as directly attached terminal users of that host.

threshold. (1) In IBM bridge programs, a value set for the maximum number of frames that are not forwarded across a bridge due to errors, before a "threshold exceeded" occurrence is counted and indicated to network management programs. (2) An initial value from which a counter is decremented to 0, or a value to which a counter is incremented or decremented from an initial value.

throughput class. In packet switching, the speed at which data terminal equipment (DTE) packets travel through the packet switching network.

time to live (TTL). A technique used by best-effort delivery protocols to inhibit endlessly looping packets. The packet is discarded if the TTL counter reaches 0.

timeout. (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (I) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) In LANs, a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

token ring. (1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations. (2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. (3) See also *local area network (LAN)*.

token-ring network. (1) A ring network that allows unidirectional data transmission between data stations, by a token passing procedure, such that the transmitted data return to the transmitting station. (T) (2) A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

topology. In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

topology database update (TDU). A message about a new or changed link or node that is broadcast among APPN network nodes to maintain the network topology database, which is fully replicated in each network node. A TDU contains information that identifies the following:

- The sending node
- The node and link characteristics of various resources in the network
- The sequence number of the most recent update for each of the resources described.

trace. (1) A record of the execution of a computer program. It exhibits the sequences in which the

instructions were executed. (A) (2) For data links, a record of the frames and bytes transmitted or received.

transceiver (transmitter-receiver). In LANs, a physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and that sense collisions.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission group (TG). (1) A connection between adjacent nodes that is identified by a transmission group number. (2) In a subarea network, a single link or a group of links between adjacent nodes. When a transmission group consists of a group of links, the links are viewed as a single logical link, and the transmission group is called a *multilink transmission group (MLTG)*. A *mixed-media multilink transmission group (MMMLTG)* is one that contains links of different medium types (for example, token-ring, switched SDLC, nonswitched SDLC, and frame-relay links). (3) In an APPN network, a single link between adjacent nodes. (4) See also *parallel transmission groups*.

transmission header (TH). Control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

transparent bridging. In LANs, a method for tying individual local area networks together through the medium access control (MAC) level. A transparent bridge stores the tables that contain MAC addresses so that frames seen by the bridge can be forwarded to another LAN if the tables indicate to do so.

transport layer. In the Open Systems Interconnection reference model, the layer that provides a reliable end-to-end data transfer service. There may be relay open systems in the path. (T) See also *Open Systems Interconnection reference model*.

trap. In the Simple Network Management Protocol (SNMP), a message sent by a managed node (agent function) to a management station to report an exception condition.

tunneling. To treat a transport network as though it were a single communication link or LAN. See also *encapsulation*.

T1. In the United States, a 1.544-Mbps public access line. It is available in twenty-four 64-Kbps channels. The European version (E1) transmits 2.048 Mbps.

U

universally administered address. In a local area network, the address permanently encoded in an adapter at the time of manufacture. All universally administered addresses are unique. Contrast with *locally administered address*.

User Datagram Protocol (UDP). In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

V

V.24. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE).

V.25. In data communication, a specification of the CCITT that defines the automatic answering equipment and parallel automatic calling equipment on the General Switched Telephone Network, including procedures for disabling of echo controlled devices for both manually and automatically established calls.

V.35. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at various data rates.

V.36. In data communication, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at rates of 48, 56, 64, or 72 kilobits per second.

VINES. VIrtual NEtworking System.

virtual circuit. (1) In packet switching, the facilities provided by a network that give the appearance to the user of an actual connection. (T) See also *data circuit*. Contrast with *physical circuit*. (2) A logical connection established between two DTEs.

virtual link. In Open Shortest Path First (OSPF), a point-to-point interface that connects border routers that are separated by a non-backbone transit area. Because area routers are part of the OSPF backbone, the virtual link connects the backbone. The virtual links ensure that the OSPF backbone does not become discontinuous.

VIrtual NEtworking System (VINES). The network operating system and network software from Banyan Systems, Inc. In a VINES network, virtual linking allows all devices and services to appear to be directly connected to each other, when they may actually be thousands of miles apart. See also *StreetTalk*.

virtual route (VR). (1) In SNA, either (a) a logical connection between two subarea nodes that is physically realized as a particular explicit route or (b) a logical connection that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual route pacing, and provides data integrity through sequence numbering of path information units (PIUs). (2) Contrast with *explicit route (ER)*. See also *path* and *route extension (REX)*.

W

wide area network (WAN). (1) A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) (2) A data communication network designed to serve an area of hundreds or thousands of miles; for example, public and private packet-switching networks, and national telephone networks. (3) Contrast with *local area network (LAN)* and *metropolitan area network (MAN)*.

wildcard character. Synonym for *pattern-matching character*.

X

X.21. An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for a general-purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operations on a public data network.

X.25. (1) An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. (2) See also *packet switching*.

Xerox Network Systems (XNS). The suite of internet protocols developed by the Xerox Corporation.

Although similar to TCP/IP protocols, XNS uses different packet formats and terminology. See also *Internetwork Packet Exchange (IPX)*.

Z

zone. In AppleTalk networks, a subset of nodes within an internet.

Zone Information Protocol (ZIP). In AppleTalk networks, a protocol that provides zone management service by maintaining a mapping of the zone names and network numbers across the internet on the session layer.

zone information table (ZIT). A listing of network numbers and their associated zone name mappings in the internet. This listing is maintained by each internet router in an AppleTalk internet.

Index

E

Tell Us What You Think!

Nways Multiprotocol Access Services
Software User's Guide
Version 1 Release 1

Publication No. SC30-3886-00

We hope you find this publication useful, readable, and technically accurate, but only you can tell us!
Your comments and suggestions will help us improve our technical publications. Please take a few
minutes to let us know what you think by completing this form. If you are in the U.S.A., you can mail this
form postage free or fax it to us at 1-800-253-3520. Elsewhere, your local IBM branch office or
representative will forward your comments or you may mail them directly to us.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific comments or problems:

_____

_____

_____

Please tell us how we can improve this book:

_____

_____

_____

Thank you for your comments. If you would like a reply, provide the necessary information below.

_____   _____
Name   Address

_____   _____
Company or Organization

_____   _____
Phone No.

Tell Us What You Think!
SC30-3886-00

IBM®
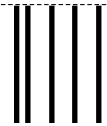
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES
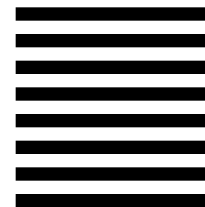
BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Design & Information Development
Dept. CGF/Bldg. 656
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

SC30-3886-00

IBM®

Part Number: 85H7918

Printed in U.S.A.

85H7918