Nways Multiprotocol Access Services

# Software User's Guide
# Version 3 Release 1

SC30-3886-03

Nways Multiprotocol Access Services

# Software User's Guide
# Version 3 Release 1

SC30-3886-03

**Note**
Before using this document, read the general information under “Notices” on page xxxi.

**Fourth Edition (June 1998)**

This edition applies to Version 3 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 non-exclusive 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 | PS/2 |
| AIX | Micro Channel | RS/6000 |
| AIXwindows | NetView | System/370 |
| APPN | Nways | VTAM |
| BookManager | ESCON | |

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names 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 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 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-C90). 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 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 pre-loaded 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, Routing, and Switched Services,* 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:

   ```
   reload
   ```

   In this example, you can enter either the whole command (reload) or its abbreviation (rel).
2. Keyword choices for a parameter are enclosed in brackets and separated by the word or. For example:

   ```
   command [keyword1 or keyword2]
   ```

   Choose one of the keywords as a value for the parameter.
3. Three periods following an option mean that you enter additional data (for example, a variable) after the option. For example:

   ```
   time 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.
4. 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.
5. Keyboard key combinations are indicated in text in the following ways:
   - **Ctrl-P**
   - **Ctrl -**
6. Names of keyboard keys are indicated like this: **Enter**
7. Variables (that is, names used to represent data that you define) are denoted by italics. For example:

   File Name: *filename.ext*

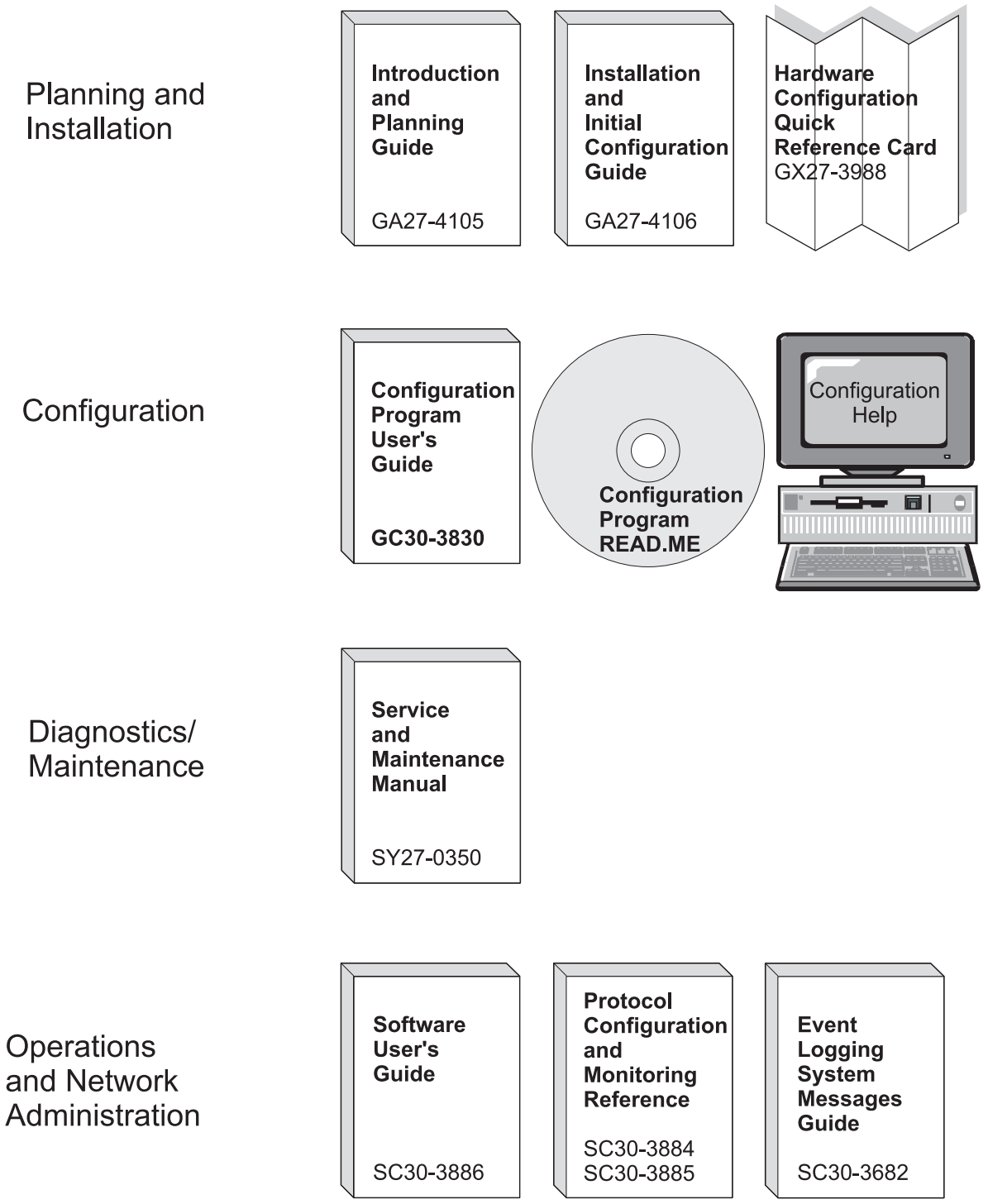



*Figure 1. Common Tasks and the IBM 2216 Library*

## 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 Introduction and Planning 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 Installation and Initial Configuration 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 Service and Maintenance Manual*

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.

## Summary of Changes for the IBM 2216 Software Library

The changes consist of:

- **New functions:**
  - Fast Token Ring (FasTR)
  - Network Address Translation (NAT) - allows a remote workstation to use a single IP address to reach different destinations behind a router.
  - Parallel Channel Support
  - Virtual Router Redundancy Protocol (VRRP) - allows a set of routers on a LAN that are running this protocol to back up each other.
  - IP, IPX and AppleTalk can now be routed on the same unit, but on separate interfaces.
- **Enhanced functions:**
  - APPN
    - Extended Border Node support
    - TN3270E subarea connectivity support
  - Base Services
    - Maximum number of network interfaces increased
    - Event Logging System (ELS) enhancements

  - BGP
    - Supports the **reset** command
  - DLSw
  - Dynamic Reconfiguration
  - FDDI - now supports Banyan VINES
  - Frame Relay - now supports encryption
  - IP
    - Security enhancements to support firewalls
    - Filtering enhancements to support security
    - IP routing on bridged network
    - Version 4 Precedence setting and filtering support for APPN/HPR, SNA/DLSw, and TN3270 Server
    - Supports the **reset** command
  - IPX
    - Supports the **reset** command
  - OSPF
    - Enhancements in support of RFC 2178
    - Supports the **reset** command
  - Security Enhancements
    - TACACS+/RADIUS Authorization and Accounting
      - You can enable TACACS+/RADIUS to control login to the router
- **Clarifications and corrections**

  The technical changes and additions are indicated by a vertical line (|) to the left of the change.

## Under Reconstruction

This edition begins a number of editorial changes to this book and the other software books that will:

- Reorganize the material
- Remove any unnecessary and redundant information
- Improve retrievability
- Add additional clarification to some information

The following information has been moved as part of this reorganization:

- **Using and configuring BGP**

  This has been moved:

  **From** *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1*

  **To** *Protocol Configuration and Monitoring Reference Volume 2 for Nways Multiprotocol Access Services Version 3 Release 1*
- **Using and configuring NHRP**

  This has been moved:

  **From** *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1*

**To** *Protocol Configuration and Monitoring Reference Volume 2 for Nways Multiprotocol Access Services Version 3 Release 1*

This effort will take a number of editions. If you would like to comment on this reorganization, please mail or fax your comments on the form for readers' comments at the back of this publication.

## Summary of Changes

# Part 1. Understanding and Using the Software

# Chapter 1. Getting Started

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

- Router console terminals
- Router software (Multiprotocol Access Services)
- Router software user interface

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

- "Before You Begin"
- "Migrating to the Current Release"
- "Accessing the Software Using Local and Remote Consoles"

## Before You Begin

Before you begin, 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 correct 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 Installation and Initial Configuration Guide*.

## Migrating to the Current Release

Refer to the *Service and Maintenance Manual* for information about migrating to a new code level.

## Accessing the Software Using Local and Remote Consoles

The router console lets you use the router user interface to monitor and change the function of the router's networking software (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

using 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.

**Important:** Garbage, random characters, reverse question marks, or the inability to connect your terminal to the 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 IBM 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 IBM 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 (ac) 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 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 Installation and Initial Configuration 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 through a Telnet connection. 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.

## 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
   ```

   At this point, it makes 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, 72. Remember to reload to activate any changes.

   **Note:** If you do not enter a login name and valid password within 1 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 asterisk (*) prompt.

   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 reenter a command. See “Command History for GWCON and CONFIG Command Line” on page 22 for information on how to access previously entered commands.

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

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

5. Exit the router as described in “Exiting the Router”.

## Reloading the Router

Whenever you change a user-configurable parameter that is not dynamically configurable, you must reload the router for the change to take effect. To do so, enter the OPCON **reload** command. For example:

```
* reload

The configuration has been changed, save it? (Yes or [No]  or Abort)

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

## 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 software (Multiprotocol Access Services) is a multitasking 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.

## Understanding the First-Level User Interface

The user interface to the software consists of the main menu (process) and several subsidiary menus (processes). These menus are related to the multiple levels of processes in the software.

The first level of processes consists of the OPCON and CONFIG-ONLY processes. In most cases, you will use the OPCON process to access the second level to configure or operate the base services, features, interfaces, and protocols you will run on your IBM 2216.

The second level of processes consists of the processes listed by the **status** command. You use the talk *pid* command to access the second-level processes. There are processes that you cannot use in the software. See Table 1 on page 10 for an overview of the processes.

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

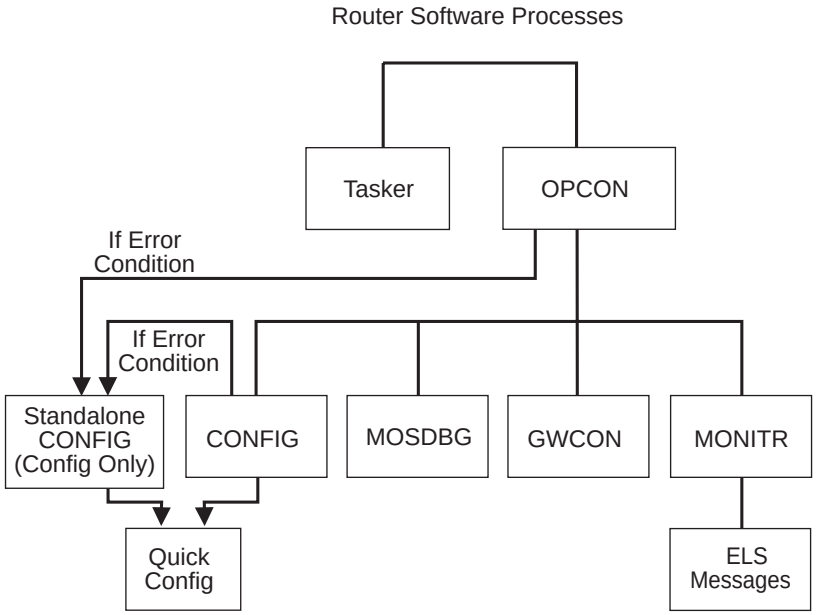



*Figure 2. Multiprotocol Access Services*

Figure 3 on page 8 is an example of the relationship between the various process levels.

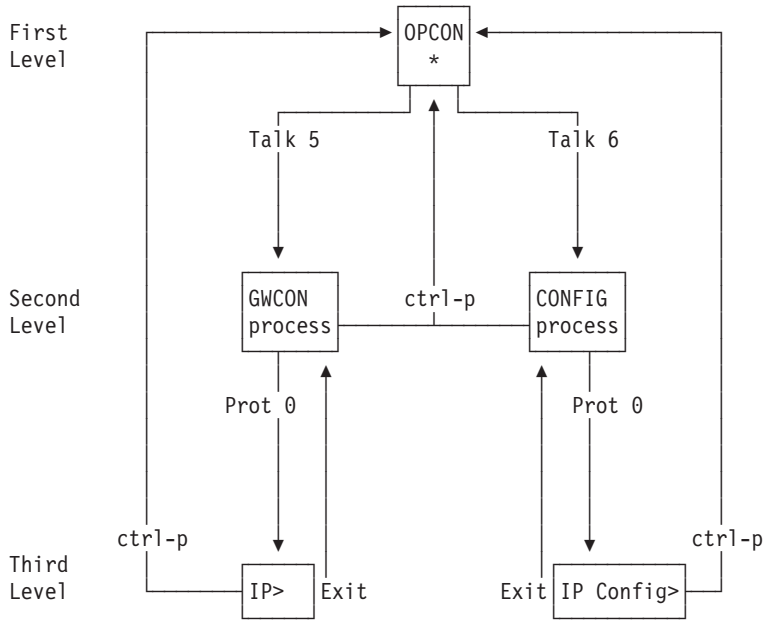



*Figure 3. Relationship of Processes and Commands*

**Note:** Also shown in Figure 3 are the various commands to access each process level and return from each process level.

See “Chapter 4. The OPCON Process and Commands” on page 29 for more information about OPCON and “Config-Only Mode” on page 58 for more information about CONFIG-ONLY.

The ROPCON process handles processing from remote consoles and is essentially the same as the OPCON process.

## 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.

## System Security

Multiple users with login permissions can also be added using the **add user** command. See “Configuring User Access” on page 60 for details on security issues and for information on the **set password** and **add user** commands.

# Chapter 2. Using the Software

This chapter describes how to use the software. It consists of:



## Entering Commands

When typing a command, remember the following:

- Type only enough sequential letters of the command to make it unique among the available commands. For example, to execute the **reload** command you must enter **rel** as a minimum. The minimum number of required characters are underlined in the command syntax chapters.
- 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.

## 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    COpCn1    RDY    TTY0
2    Monitr    DET     --
3    Tasker    RDY     --
4    MOSDDT    DET     --
5    CGWCon    DET     --
6    Config    DET     --
7    ROpCn1    IDL    TTY1 128.185.210.125
8    ROpCn2    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 4. The OPCON Process and Commands” on page 29 .)

   **Note:** Not every processes listed has a user interface (for example, the **talk 3** process). The **talk 4** command is for use by the IBM service representatives.

## 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 five main processes:

*Table 1. Processes, Their Purpose, and Commands to Access*

| Process | Level and Purpose | Command to Access | Input Prompt |
|---|---|---|---|
| OPCON | Level 1 - access to all secondary levels | **Ctrl-P** | asterisk (*) |
| CONFIG | Level 2 - base services configuration and access to configuration third level | **talk 6** | Config > |
| GWCON | Level 2 - base services operation and monitoring and access to operations and monitoring on third level | **talk 5** | plus sign (+) |
| MONITR | level 2 - message display | **talk 2** | (none) |
| MOSDDT | level 2 - diagnostic environment | **talk 4** | $ |
| **Note:** Only enter the **talk 4** command under the direction of a service representative. | | | |

At the OPCON 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 reenter a command. See “Command History for GWCON and CONFIG Command Line” on page 22 for information on how to access previously entered commands.

## 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 level. To do this, type **?** (the **help** command), and then press **Enter**. Use **?** to list the commands that are available from the current level. You can usually enter a **?** after a specific command name to list its options. For example, the following information 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
```

## Exiting a Lower Level Environment

The multiple-level nature of the software places you in secondary, tertiary, and even lower level environments as you configure or operate the 2216. To return to the next higher level, enter the **exit** command. To get to the secondary level, continue entering **exit** until you receive the secondary level prompt (either `Config>` or +).

For example, to exit the IP protocol configuration process:

```
IP config> exit
Config>
```

If you need to get to the primary level (OPCON), enter the intercept character (**Ctrl P** by default).

## 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 the default *intercept character*.

If you use the intercept character (the default intercept character is **Ctrl-P**) from a third-level or lower level process to return to the * prompt, the next time you use the **talk** command, you will reenter the third level process. This link goes away when the router is re-initialized.

# Some Configuration Suggestions

Configuring a 2216 is different depending on whether you are configuring for the first time, creating a configuration based on an existing configuration, or just updating a configuration. Use the following sections as a guide to the best procedure to use, depending on your needs.

## Creating a First Configuration

This procedure assumes that you have no other 2216 that contains a configuration similar to the one for the 2216 you are configuring. The procedure also assumes that you have just taken the 2216 out of the box. Although this procedure specifies an order, you can perform the actual configuration (after step 3) in any order.

To configure a IBM 2216 for the first time:

1. Examine the 2216 you are configuring to determine what interfaces you need to configure. Note these for later use.
2. Connect to the 2216 as described in “Accessing the Software Using Local and Remote Consoles” on page 3.

3. Initially configure a port on the 2216 and at least an internal IP address for the device using quick config as described in “Quick Configuration” on page 59 or “Appendix A. Quick Configuration Reference” on page 909. Configure the minimum needed to allow you to Telnet into the device.
4. Configure any base services, such as boot options. Access the configuration process as described in “Accessing the Configuration Process, CONFIG (Talk 6)” on page 14.
5. Configure the interfaces. Access the interface configuration process as described in “Accessing the Network Interface Configuration Process” on page 15 .
6. Configure any required features. Access the feature configuration process as described in “Accessing Feature Configuration and Operating Processes” on page 20 .
7. Configure any protocols that will run through this device. Access the protocol configuration process as described in “Accessing Protocol Configuration and Operating Processes” on page 21.

   **Note:** At the very least, you will configure IP in this step.
8. Reload the router as described in “Reloading the Router” on page 6.

# Basing a Configuration on an Existing Configuration

This section describes how to:

- Base a configuration on the configuration in an operating 2216
- Permanently update the configuration in a 2216
- Temporarily updating the configuration of a 2216 while the 2216 is operating

## Basing on an Existing Configuration

If you already have a 2216 that has the same interfaces, features, and protocols that you will configure on a new 2216, you can save time during configuration by basing the configuration on the existing 2216. You can perform this type of configuration either using the command line interface or by using the configuration program that comes with the 2216. In both cases, the procedures assume that the 2216 is not in your production network.

To base a configuration on an existing configuration using the command line interface:

1. Obtain a copy of the configuration you’ll be using.
   a. Enter **talk 6** at the OPCON (*) prompt.
   b. Enter **boot** at the `Config>` prompt.
   c. Enter the **copy configuration** *file* at the `Boot config>` prompt. See “Chapter 6. Using BOOT Config to Perform Change Management” on page 41 for more information.
2. Connect to the 2216 that you are configuring.
3. Load the configuration you obtained in step 1 into the 2216 using TFTP. See “Chapter 6. Using BOOT Config to Perform Change Management” on page 41.
4. Update the configuration.
5. Write the configuration. See “Chapter 8. The Configuration (CONFIG) Process and Commands (Talk 6)” on page 57.

6. Reload the 2216.

To base a configuration on an existing configuration using the configuration program:

1. Start the configuration program.
2. Retrieve the configuration from the 2216 on which you want to base this configuration.
3. Make the changes you need for the new configuration. These changes include addresses, the host names, users, and other items.
4. Save the configuration with a different name from the name that you used to retrieve the configuration.
5. Send the configuration to the 2216 you are configuring.
6. Reload the 2216.

For more about using the configuration program, see *Configuration Program User's Guide for Nways Multiprotocol Access, Routing, and Switched Services,* GC30-3830.

## Permanently Updating a Configuration

To permanently update a configuration:

1. Access the 2216 you are updating as described in "Accessing the Software Using Local and Remote Consoles" on page 3. You should see the * prompt.
2. Enter the **talk 6** command to access the configuration process.
3. Enter the appropriate commands to access the third-level process that configures the areas that you are changing.
4. Enter **exit** as many times as needed to return to the configuration process.
5. Write the configuration. See "Chapter 8. The Configuration (CONFIG) Process and Commands (Talk 6)" on page 57.
6. Reload the 2216.

## Temporarily Updating a Configuration

The ability to temporarily update a configuration allows you to make changes to some of the operating characteristics of a 2216 until such time that you can make permanent updates to the configuration. This enables you to implement changes immediately that would resolve problems or improve performance and avoid an outage during a peak period. You can then make permanent updates to the configuration and schedule an outage so you can reload to pick up the change.

To temporarily update a configuration:

1. Access the 2216 you are updating as described in "Accessing the Software Using Local and Remote Consoles" on page 3. You should see the * prompt.
2. Enter the **talk 5** command to access the operating/monitoring process.
3. Enter the appropriate commands to access the third-level process that monitors the areas that you are changing.
4. Enter **exit** as many times as needed to return to the operating/monitoring process.
5. Enter **Ctrl-P** to return to the * prompt.
6. Exit the router as described in "Exiting the Router" on page 6

## Accessing the Second-Level Processes

All interfaces, features, and protocols have commands that you use to access the following processes:

- The *configuration* process to initially configure and enable the interface, feature, or protocol, as well as perform later configuration changes.
- The operating/monitoring process to display information about each interface, feature, or protocol, to make temporary configuration changes, or to activate configuration changes.

You can also configure or operate some base system services through the second-level processes. The commands to perform these functions are described starting in “Chapter 8. The Configuration (CONFIG) Process and Commands (Talk 6)” on page 57 and “Chapter 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands” on page 99.

The next sections describe the procedures for accessing the second-level processes.

## Accessing the Configuration Process, CONFIG (Talk 6)

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>`.

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, 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 (see “Quick Configuration” on page 59).

### Reloading the Router

Changes that you make to the protocol parameters through CONFIG do not take effect until you either activate the net that contains any dynamic changes or reload the router software.

**Note:** You must enter the **write** command to save the changes in the device's flash memory.

## Accessing the Operating/Monitoring Process, GWCON (Talk 5)

To view information about the interfaces, features, or protocols or to change parameters while running, you must access and use the operating (monitoring) process. Operating command interfaces are modes of the GWCON interface. Within the GWCON mode, each interface, feature, or protocol interface has its own prompt. For example, the prompt for the TCP/IP protocol is `IP>`.

**Note:** Any parameters you change in this process will not remain active across any event that causes the 2216 to reload the operational code, such as a power outage or entering the **reload** command.

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, enter the **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.

## Accessing the Third-Level Processes

After accessing the second level, you will need to enter commands on the third level to configure or operate the interfaces, features, and protocols in your IBM 2216. The following sections describe how to access the third level processes.

## Accessing Network Interface Configuration and Operating Processes

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

### 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 OPCON **talk** command and the PID for CONFIG. (For more details about this command, refer to "Chapter 4. The OPCON Process and Commands" on page 29.)

   ```
   * 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 (Enter **add device ?** to get a list of the supported device types):

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.

The 4-port ISDN Channelized T1 and E1 adapters allow you to configure multiple ports with one **add device** command. When you add one of these adapters, the software will prompt you for a range of ports to add. The following example shows how you would add a 4-port ISDN Channelized T1 and E1 adapter into slot 4 that is using dial-in nets:

```
Device Slot #(1-8) [1]? 4
Device Port Range (1-8)
   Lowest  Port #(1) [1]? 2
   Highest Port #(8) [8]? 2
Automatically add dial-in nets for this base net? (Yes or [No]): yes
Automatically enable IP for these dial-in nets?(Yes or [No]): yes
Enable as a Multilink PPP link?(Yes or [No]): yes
Adding 23 dial-in nets on top of base net 14
Adding 8-port ISDN Primary T1/J1 devices in slot 4 port 2 as interfaces #14.
Use "net 14" to configure 8-port ISDN Primary T1/J1 parameters.
```

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
Enter the number of PPP Dial Circuit interfaces [1]?
Adding device as interface 8
Base net for this circuit[0]?4
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
```

d. The following example adds a dial-in circuit.

```
Config>add device dial-in
Enter the number of dial-in interfaces [1]?
Adding device as interface 5
Base net for this circuit [0]? 5
Defaulting Data-link protocol to PPP
Use "set data-link" command to change the data-link protocol
Use "net 5" command to configure circuit parameters
```

e. Multilink PPP

The following example adds a multilink PPP interface:

```
Config>add device multilink-ppp
Enter the number of Multilink PPP interfaces [1]?
Adding device as interface 7
Defaulting Data-link protocol to PPP
Use "net 7" command to configure circuit parameters
```

**Notes:**

a. 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 2 on page 18 for the data-link types supported on serial ports and dial circuits, and to the description of the **set data-link** command on page 93.

2. 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
```

3. Record the interface numbers.
4. 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
```

***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 eight ISDN-PRI interfaces may be defined with the following restrictions:
  - No more than four 1-port adapters may be used

– No more than one 4-port ISDN Channelized T1 and E1 adapter may be used

- 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 fan out cable attached to the V.35/V.36 adapter determines which type of cables (V.35 or V.36) can be used.

***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 IBM 2216's network interfaces.

Table 2 lists network architectures and the supported interfaces for each architecture.

*Table 2. Network Architecture and the Supported Interfaces*

| Network Architecture | Supported Interfaces |
|---|---|
| ATM | 1-Port ATM -Mbps MMF<br><br>1-Port ATM -Mbps SMF |
| 802.5 Token-Ring | 2-Port Token-Ring |
| Ethernet | • 2-Port 10-Mbps Ethernet<br>• 1-port 10/100-Mbps Ethernet |
| ISDN | **Note:** The interfaces marked with an asterisk (*) can be used either as ISDN or channelized interfaces.<br><br>1-Port ISDN-PRI (T1/J1) *<br><br>1-Port ISDN-PRI (E1) *<br><br>4-port ISDN Channelized T1 and E1 * |
| Point-to-Point | 8-port V.24/EIA 232E, 6-port V.35/V.36, 8-port X.21, 1-port HSSI, and dial circuit interfaces |
| Frame Relay | 8-port V.24/EIA 232E, 6-port V.35/V.36, 8-port X.21, 1-port HSSI, and dial circuit interfaces |
| X.25 | 8-port V.24/EIA 232E, 6-port V.35/V.36, and 8-port X.21 and dial circuits |
| SDLC Relay | and 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, 8-port X.21, and dial circuit interfaces |

*Table 2. Network Architecture and the Supported Interfaces (continued)*

| Network Architecture | Supported Interfaces |
|---|---|
| V.25bis | 8-port V.24/EIA 232E |
| Multilink PPP (MP) | Not supported on physical interfaces, only on a ISDN virtual interface |
| ESCON Channel Adapter | LAN Channel Station (LCS), Link Services Architecture (LSA), and Multi-Path Channel+ (MPC+) |
| Parallel Channel Adapter | LAN Channel Station (LCS), Link Services Architecture (LSA), and Multi-Path Channel+ (MPC+) |
| FDDI | 1-port FDDI |

**Notes:**

1. PPP dial circuit interfaces can use an ISDN, or V.25bis as the base network interface.
2. FR dial circuit interfaces can use an ISDN or a V.25bis network as the base network interface.
3. Dial-In circuit interfaces can use an ISDN network as the base network interface.
4. SDLC dial circuits use V.25bis as the base network interface.
5. X.25 can use ISDN B-channels as the base network interface.

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

   ```
   * talk 5
   ```

2. 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

   Multiprotocol Access Services

   5765-D47 Feature 2802 V3 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
```

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

```
+ network 11
X.25>
```

   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.

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

## Accessing Feature Configuration and Operating Processes

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

### Accessing the Feature Processes

Use the **feature** command from the CONFIG process to access configuration commands for specific 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 operating 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 8 on page 84 lists the available feature numbers and names.

Once you access the configuration or operating 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 Protocol Configuration and Operating Processes

This section describes how to access the protocol configuration and operating processes.

## Entering a 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 page 85 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.

## Entering a Protocol Operating 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-D47 Feature 2802 V3 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.

## Command History for GWCON and CONFIG Command Line

The Command History contains up to the last 50 commands entered by the user in GWCON (Talk 5) or CONFIG (Talk 6) command line menus.

Backward and Forward retrieve keys can be used to recall previously entered commands. In addition, a facility is provided to enable the advanced user to repeat a particular series of commands.

## Repeating a Command in the Command History

By hitting **Ctrl-B** (backward) or **Ctrl-F** (forward) at any command line prompt in a GWCON or CONFIG menu, the current command line is replaced by the previous or next command in the Command History. The Command History is common to both GWCON and CONFIG. That is, a command entered while in a GWCON menu can be retrieved from within CONFIG and a command entered while in a CONFIG menu can be retrieved from within GWCON.

The Command History contains the most recently entered commands, up to a maximum of the last 50 commands. If only three commands have been entered since a restart, pressing **Ctrl-F** or **Ctrl-B** circles through only those three commands. If no commands have been entered thus far, **Ctrl-F** or **Ctrl-B** results in a “bell”, the same bell you see when trying to backspace beyond the beginning of a line of text.

**Note:** A command aborted by pressing **Ctrl-U** will not be entered into the Command History.

To enter two similar commands:

```
display sub les
display sub lec
```

Enter:

**`display sub les`**, then press **Enter**

**Ctrl-B** for Backward, and the current line is replaced with-

**`display sub les`**

Press **Backspace** and replace “s” with “c” to get

**`display sub lec`** and then press **Enter**

## Repeating a Series of Commands in the Command History

There is an additional feature for advanced users to facilitate repeating a particular series of GWCON or CONFIG commands. C1, C2,...,Cn in the Command History is referred to as a *repeat sequence*. This feature may be more convenient than simply using **Ctrl-B** and **Ctrl-F** when you must repeat a given task that requires multiple commands. Enter **Ctrl-R** (repeat) to set the start of the *repeat sequence* at command C1. Enter **Ctrl-N** (next) successively to retrieve the next command(s) in the repeat sequence. Commands are not automatically entered, but are placed on the current command line allowing you to modify or enter the command.

To produce the desired behavior of a repeat sequence, the first command retrieved using the first **Ctrl-N** (next) depends on the manner in which the start of the repeat sequence was set using **Ctrl-R** (repeat).

Setting the start of the repeat sequence with **Ctrl-R** can be done in two ways:

1. When C1 is initially entered
2. When C1 is retrieved from the Command History with **Ctrl-B** or **Ctrl-F**.

## Starting a Repeat Sequence As Commands Are Entered

If you enter **Ctrl-R** as command C1 is being keyed in, and then enter commands C2, C3... Cn. **Ctrl-N** will successively bring commands C1, C2, ... Cn, C1, C2, ... Cn, C1, ... to the command line.

In Example 1, the start of the repeat sequence is set as the first command is keyed in. The user knows ahead of time that the same commands to be entered in GWCON need to be repeated in CONFIG.

### Example 1

1. As the first command in the sequence is keyed in, use **Ctrl-R** (repeat) to set the start of the repeat sequence.

   ```
   *talk 5
   +event Ctrl-R
   ```

   then press **Enter** to set the start of the repeat sequence.
2. Continue typing the subsequent commands in the sequence:

   ```
   Event Logging System user console
     ELS>display sub les
     ELS>display sub lec
     ELS>exit
     +
   ```
3. To enter these same commands in CONFIG, press

   Ctrl-P (the default OPCON intercept character) and go to CONFIG.

   ```
   +-press Ctrl-P-
   *talk 6
   Config>Ctrl-N for NEXT to retrieve the start of
           this sequence-
   Config>event Enter
   Event Logging System user configuration
   ELS config>Ctrl-N for NEXT to retrieve the next
              command in sequence-
   ELS config>display sub les Enter
   ELS config>Ctrl-N for NEXT to retrieve the next
              command in sequence-
   ELS config>display sub lec Enter
   ELS config>Ctrl-N for NEXT to retrieve the next
              command in sequence-
   ELS config>exit Enter
   Config>
   ```

## Starting a Repeat Sequence After All Commands Are Entered

On the other hand, if you first enter C1, C2, ... Cn, and retrieve C1 via **Ctrl-B** or **Ctrl-F**. Entering **Ctrl-R**, entering **Ctrl-N** successively brings commands C2,..., Cn, C1, C2,..., Cn, C1,...,Cn to the command line (see Example 2). The first occurrence of C1 is bypassed since C1 is already available on the command line at the time it was retrieved, and does not need to be recalled again by the first **Ctrl-N**.

In Example 2, all the commands are entered and then the first command in the sequence to be repeated is retrieved. A sequence of commands has been entered in GWCON, and the same sequence needs to be repeated in CONFIG.

### Example 2

1. Enter the following commands in GWCON:

```
*talk 5
+event
Event Logging System user console
ELS>display sub les
ELS>display sub lec
ELS>exit
+
```

2. To enter these same commands in CONFIG, press **Ctrl-P** (the default OPCON intercept character) and go to CONFIG.

```
+Ctrl-P-
*talk 6
Config>Ctrl-B four times to retrieve the start of
        the four command sequence in this example-
Config>event
Config>event Ctrl-R for REPEAT to set the start of
             the repeat sequence-
 Config>event Enter
 Event Logging System user configuration
 ELS config>Ctrl-N for NEXT to retrieve the next
             command in sequence-
 ELS config>display sub les Enter
 ELS config>Ctrl-N for NEXT to retrieve the next
             command in sequence-
 ELS config>display sub lec Enter
 ELS config>Ctrl-N for NEXT to retrieve the next
             command in sequence-
 ELS config>exit Enter
 Config>
```

If the OPCON **intercept** command described in “Chapter 4. The OPCON Process and Commands” on page 29 has been used to redefine the OPCON intercept character from the default character **Ctrl-P** to one of the Command History control characters, **Ctrl-B**, **Ctrl-F**, **Ctrl-R**, or **Ctrl-N**, the OPCON intercept character will take priority. For example, if the intercept character has been changed to **Ctrl-F**, then **Ctrl-F** will not retrieve Forward in the Command History, but will instead place the user back at the OPCON prompt (*).

# Chapter 3. 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 Installation and Initial Configuration 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 router, note that:

- 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 48 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.

**Important:** In release 2, multiple load modules make up a single device load. If you are transferring a load into a bank using XMODEM, you must transfer the files individually. The following applies:

1. Transfer LML.ld first
2. You must ensure all files that make up the load transfer successfully. When there is an error transferring a file, you will receive a message box containing “ERROR WRITING FILE”. Otherwise, you can assume the file has transferred successfully.
3. When all files have transferred, the status of the bank will change from “Corrupt” to “Avail”.

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 4. The OPCON Process and Commands

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



## 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. 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)
- Return to the Base LAN Switch console
- 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

# Chapter 5. Configuring OPCON

This chapter describes the OPCON interface configuration and operational commands. It includes the following sections:

- "Accessing the OPCON Process"
- "OPCON Commands"

## Accessing the OPCON Process

When the router starts for the first time, a boot message appears on the console. Then the OPCON prompt (*) appears on the console, indicating that the OPCON process is active and ready to accept commands.

The OPCON process allows you to configure, change, 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 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 3. To use them, access the OPCON process and enter the appropriate command at the OPCON prompt (*).

*Table 3. OPCON Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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. |

*Table 3. OPCON Commands (continued)*

| Command | Function |
|---|---|
| Talk | Connects to another router process and enables the use of its commands. |
| Telnet | Connects to another router. |

## 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:**

**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 described in “Halt” on page 33.

**Syntax:**

**divert** *pid tty#*

**Example: `divert 2 1`**

```
Copyright Notices:
Copyright IBM Corp. 1994, 1997
MOS Operator Control


* divert 2 1

* status
Pid  Name      Status TTY  Comments
1    COpCN1    IOW    TTY0 gzs
2    Monitr    IDL    TTY0
3    Tasker    RDY     --
4    MOSDBG    DET     --
5    CGWCon    DET     --
```

```
6    Config    DET     --
7    ROpCN1    IDL    TTY1
8    ROpCN2    RDY    TTY2 jlg@128.185.40.40
```

## 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. 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:**

**flush** *pid*

**Example:** `flush 2`

```
* status
Pid  Name      Status TTY  Comments
1    COpCN1    IOW    TTY0 gzs
2    Monitr    IDL    SNK
3    Tasker    RDY     --
4    MOSDBG    DET     --
5    CGWCon    DET     --
6    Config    DET     --
7    ROpCN1    IDL    TTY1
8    ROpCN2    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. 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`

```
* status
Pid  Name      Status TTY  Comments
1    COpCN1    IOW    TTY0 gzs
2    Monitr    IDL     --
3    Tasker    RDY     --
4    MOSDBG    DET     --
5    CGWCon    DET     --
6    Config    DET     --
7    ROpCN1    IDL    TTY1
8    ROpCN2    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. If you change the OPCON intercept character from the default, **Ctrl-P**, to one of the Command History control characters, **Ctrl-B**, **Ctrl-F**, **Ctrl-R**, or **Ctrl-N**, the OPCON intercept character will take priority.

For example, if you change the intercept character to **Ctrl-F**, then **Ctrl-F** will not retrieve Forward in the Command History, but will instead return to the OPCON prompt (*). See “Command History for GWCON and CONFIG Command Line” on page 22 for information on how to access previously entered GWCON or CONFIG commands.

**Syntax:**

**intercept** *character*

**Example:** `intercept ^u`

From this example, the intercept character is now **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:**

**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 4 on page 35 .

**Syntax:**

**memory**

**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.

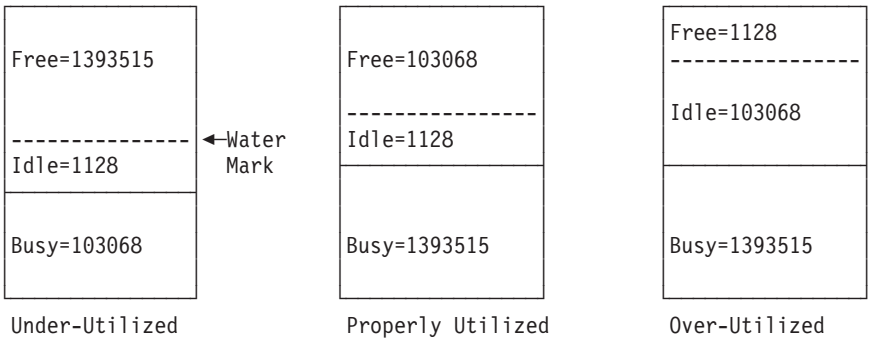



*Figure 4. Memory Utilization*

## 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:**

**reload**

**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:**

**status** *pid*

**Example: `status`**

```
Pid  Name      Status TTY  Comments

1    COpCN1    IOW    TTY0
2    Monitr    IDL     --
3    Tasker    RDY     --
4    MOSDBG    DET     --
5    CGWCon    IOW     --
6    Config    IOW    TTY1
7    ROpCN1    IOW    TTY1 128.185.46.101
8    ROpCN2    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.

**TTY***n* 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. See "Command History for GWCON and CONFIG Command Line" on page 22 for information on how to

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:**

**talk** *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 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, enable Host Services.

**Syntax:**

**telnet** *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 '^]'
```

When 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:** When using a VT100 terminal, do not press ← **Backspace** because it inserts invisible characters. 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 reenter 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
*
```

# Part 2. Understanding, Configuring, and Using Base Services

# Chapter 6. Using BOOT Config to Perform Change Management

This chapter describes how to use the Boot/Dump Configuration process. This chapter includes the following sections:

- "Understanding Change Management"
- "Using the Trivial File Transfer Protocol (TFTP)"

## Understanding Change Management

Change management is the handling of software and configuration data for an IBM 2216. This involves:

1. Moving code and configuration data to and from the IBM 2216
2. Moving code and configuration data on the IBM 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 by entering the **boot** command at the `Boot config>` prompt (talk 6), or the firmware box should be in a condition where the hard drive does not contain viable software (that is, you cannot access talk 6).

The IBM 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 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 IBM 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 an IBM 2216's non-volatile configuration memory, image bank, and remote hosts.

TFTP allows you to:

- Get a configuration file from a server to an IBM 2216
- Put a configuration file from an IBM 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 IBM 2216 acts as a client node by generating TFTP requests from the IBM 2216 console using the `Boot config>` process **tftp** command.

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

### Using BOOT Config

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

## Loading an Image at a Specific Time

There may be occasions when you may want to load a device on a specific day and time when you will be unavailable. You can configure the device to perform a timed load using the **timedload activate command**. Other commands allow you to view a device's scheduled load information or cancel a scheduled load. See "Change Management Configuration Commands" on page 43 for information on these commands.

# Chapter 7. Configuring Change Management

This chapter describe the Change management configuration commands. It includes the following sections:

- "Accessing the Change Management Configuration Environment"
- "Change Management Configuration Commands"

## Accessing the 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**.

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 summarizes the Change Management Configuration commands.

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

*Table 4. Change Management Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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. |
| Disable | Turns off various change management functions. |
| Enable | Turns on various change management functions. |
| Erase | Erases a stored image or a configuration file. |
| List | Displays information about configuration files and scheduled load information. |
| 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 the IBM 2216 and remote servers. |
| Timedload | Schedules a load into the device on a specific day and time, cancels a scheduled load, or displays scheduled load information. |
| Unlock | Removes the lock from a configuration allowing the configuration to be updated by the device. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Add

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

**Syntax:**

**add** configuration *file description*

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

  Auto-boot mode is enabled. Fast-boot mode is disabled.
```

## Copy

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

**Syntax:**

**copy** 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 bank currently in use by the IBM 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 48) the configurations, CORRUPT appears next to the bank that is damaged.

## Describe

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

**Syntax: describe**

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

```
+------------------------+------------------------+
|         BANK A         |         BANK B         |
| Product ID -   5765-D47| Product ID -   5765-D47|
| Version   3 Release   1| Version   3 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|
+------------------------+------------------------+
```

## Disable

Use the **disable** command to turn off various change management functions.

**Syntax:**

**disable** auto-boot

fast-boot

**auto-boot**

Disabling auto-boot causes the router boot sequence to halt at the firmware main menu, without running the router operational code. Disabling this function is similar to selecting “attended mode” from the firmware menu, and is useful for accessing the firmware when you are remotely dialed-in to the router console through a modem. The default auto-boot mode is “enabled”.

**Example:**

```
Boot config>disable auto-boot
Auto-boot mode is now disabled
```

**fast-boot**

Disabling fast-boot causes the router to run diagnostic tests when the router is booting during a power-on or a software reload. This provides better hardware error detection but results in slower boot times. This is the default mode, and is recommended whenever the router is in a production environment.

## Enable

Use the **enable** command to turn on various change management functions.

**Syntax:**

**enable** auto-boot

fast-boot

**auto-boot**

Enabling auto-boot causes the router boot to the router operational code without stopping at the firmware main menus. Enabling this function is similar to selecting “unattended mode” from the firmware menus, and is the default operational mode.

**Note:** To enable auto-boot mode using this command, you must also have unattended mode selected in the firmware.

**fast-boot**

Enabling fast-boot causes the router to skip diagnostic tests when the router is booting during a power-on or a software reload. This reduces

hardware error detection but results in faster boot times. The default mode is “disabled,” which is recommended whenever the router is in a production environment.

**Example:**

```
Boot config>enable fast-boot
Fast-boot mode is now enabled
```

## Erase

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

**Syntax:**.

**erase** configuration [file]

load [image]

**config or load**

Erases a configuration file or a load image. Enter the config number to be erased 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

  Auto-boot mode is enabled.  Fast-boot mode is disabled.
```

**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

  Auto-boot mode is enabled.  Fast-boot mode is disabled.
```

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. This command may also be used to display boot options and scheduled load information.

**Syntax:**

**list**

**Example:** Boot config>**list**

```
+------ 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 - 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
  Auto-boot mode is enabled.  Fast-boot mode is disabled.
```

```
 Time Activated Load Schedule Information...

The router is scheduled to reload as follows.

Date:  June 26, 1997
Time:  16:30
The load modules are in bank A.
The configuration is CONFIG 1 in bank A.
Boot config>
```

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

Auto-boot mode is enabled.  Fast-boot mode is disabled.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

  Auto-boot mode is enabled.  Fast-boot mode is disabled.
```

**Note:** Note that bank A config 4 is marked with an “L.”

## SET

Use the **set** command to select the code bank, 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:**

**set**

**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

   Auto-boot mode is enabled.  Fast-boot mode is disabled.
```

## TFTP

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

**Note:** When you unzip a Release 2 image, you will see multiple files ending in ".ld". Use the **tftp get load modules** command to get multiple load modules.

**Syntax:**

| | |
|---|---|
| **tftp get** | config |
| | load single *image* |
| | load modules |
| **tftp put** | config |
| | load single *image* |
| | load modules |

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

```
+------ 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: : (/u/bin) /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.
```

**Note for Dynamic Loading of Software:** All of the load modules in the specified directory will be retrieved as part of the load going into the bank. For loads for releases prior to Version 1, Release 2, this will be a single load module. For Version 1, Release 2 loads and later, this may be multiple load modules.

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

```
+------ 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 modules directory: : (/u/bin) /usr/2216load/
Select the destination bank:  (A, B): [A] a
  TFTP SW load image
  get:   /usr/2216load/LML.ld
  from:  192.9.200.1
  to:    bank A.

Operation completed successfully.
```

**Notes:**
When putting files to a server:

1. Make sure that the files on the target server have the appropriate permissions that would allow anyone to write to those files. If not, the put operation will fail.
2. You must be aware of the files you are putting to the target server.A To determine whether the image in the bank is a single module or multiple modules, use the **describe** command. A load prior to Version 1, Release 2 is a single module. Loads at Version 1, Release 2 or after are multiple modules.

## Timedload

Use the **timedload** command to schedule a load on a device, cancel a scheduled load, or view scheduled load information.

This command allows you to load the device outside peak network traffic periods when support personnel may not be present.

**Syntax:**

**timedload** activate
deactivate
view

**activate**

Schedules a load on the device. You will be prompted for information for a time-activated load similar to the **tftp get load** and **tftp get config** commands. See “TFTP” on page 51 for information about the parameters.

**Time of day to load the device**

Specifies the date and time to load the device. Specify the value as *YYYYMMDDHHMM*, where:

*YYYY* is the four-digit year

**Note:** If the current month on the device is December, the year data must be the current year or the following year. Otherwise, if the current month on the device is January through November, the year data must be the current year.

*MM* is the two digit month.

**MM Valid Values:** 01 to 12 with 01 representing January.

*DD* is the two-digit day of the month.

**DD Valid Values:** 01 to 31, depending on the value of MM.

*HH* is the two-digit hour in 24-hour time.

**HH Valid Values:** 00 to 23

*MM* is the two-digit minute of the hour.

**MM Valid Values:** 00 to 59

The following are examples of scheduling a load from different sources.

### Example 1. Load modules and configuration source is a remote host:

```
Boot config>timedload activate

+------ 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 - 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


Time Activated Load Processing...

Select the bank to use:  (A, B): [A] a
Do you want to put load modules into the bank? (Yes, No, Quit): [Yes] yes

Do you want to retrieve a SINGLE image or a set of MODULES? [MODULES]? modules
Specify the server IP address (dotted decimal): : [1.2.3.4] 192.9.200.1
Specify the remote modules directory: : (/u/bin) /usr/601bin/205img
The destination bank is bank A
TFTP SW load image
  get:   /usr/601bin/205img/
  from:  192.9.200.1
  to:    bank A.
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'
tftp: connect to '192.9.200.1'

Operation completed successfully.

Do you want to put a configuration into the bank? (Yes, No, Quit): [Yes] yes

Specify the server IP address (dotted decimal): : [1.2.3.4] 192.9.200.1
Specify the remote file name: : (config.dat) /tftpboot/192.9.200.6.config
The destination bank is bank A
Select the destination configuration:  (1, 2, 3, 4): [1] 1
TFTP SW configuration file
  get:   /tftpboot/192.9.200.6.config
  from:  192.9.200.1
  to:    bank A, configuration 1.
tftp: connect to '192.9.200.1'

Operation completed successfully.

Time of day to load the router (YYYYMMDDHHMM) []? 199706261630
The load timer has been activated.
Boot config>
```

### Example 2. Load modules and configuration source is a bank:

```
Boot config>timedload activate

+------ 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 - 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

Time Activated Load Processing...

Select the bank to use:  (A, B): [A] a
Do you want to put load modules into the bank? (Yes, No, Quit): [Yes] no

Do you want to put a configuration into the bank? (Yes, No, Quit): [Yes] no

Select the configuration to use:  (1, 2, 3, 4): [1] 1

Time of day to load the router (YYYYMMDDHHMM) []? 199706261630
The load timer has been activated.
Boot config>
```

**deactivate**

Cancels a scheduled load.

**Example 1: Deactivate the time activated load**

```
Boot config>timedload deactivate
Deactivate Load Timer Processing...

Do you want to deactivate the load timer? (Yes, No, Quit): [No] yes
The load timer has been deactivated.
Boot config>
```

**view** Displays scheduled load information.

```
Boot Config> timedload view
Time Activated Load Schedule Information...

The router is scheduled to reload as follows.

Date:  June 26, 1997
Time:  16:30
The load modules are in bank A.
The configuration is CONFIG 1 in bank A.
Boot config>
```

## 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

  Auto-boot mode is enabled.  Fast-boot mode is disabled.
```

**Note:** Note that bank A config 4 is no longer marked with an “L.”

# Chapter 8. The Configuration (CONFIG) Process and Commands (Talk 6)

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



## 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
- Communicate with third-level processes, including protocol environments

**Note:** Refer to the chapter entitled "Migrating to a New Code Level" in the Maintenance Guide for information about migrating to a new code level.

CONFIG lets you display or change the configuration information stored in the router's nonvolatile 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 4. The OPCON Process and Commands" on page 29).

**Note:** You must enter the **write** command to save the changes in the device's flash memory.

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 6 on page 67 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, fail, or detect a bug. 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, take any one of the following actions:

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

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 Configuraton.

See “Chapter 3. Accessing the Firmware from the Command Line Interface” on page 27 for more detail.

# 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. You can also configure an SNMP community with WRITE_READ_TRAP access. This is useful during initial setup because the configuration program uses SNMP SET commands to transfer the configuration.

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

Table 5 lists what Quick Config supports.

*Table 5. Quick Config Capabilities*

| ATM Protocols | Bridging Protocols | Routing Protocols |
|---|---|---|
| LAN Emulation | STB, SRT, SRB | 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 Quick Config.

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, see "Appendix A. Quick Configuration Reference" on page 909.

## 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.

# Configuring User Access

The router configuration process allows for a maximum of 50 user names, passwords, and levels of permission. Each user 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.

## 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 is granted the same access privileges that you have as system administrator.

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 "Change" on page 73.

# Configuring Spare Interfaces

Occasionally, you may need to configure a new interface along with its bridging and routing protocols without having to restart the device. You can accomplish this by configuring a number of ***spare interfaces*** on your device. Spare interfaces are useful when:

- You are "hot-plugging" a new adapter into your device.

  You can install the adapter, configure it and then activate it without unplugging or restarting the device.
- You are adding dial circuits to your device.

  Use spare interfaces to add new V.25bis or ISDN dial circuits on an existing V.25bis or ISDN interface.
- You are adding ATM LAN Emulation clients.

  Use spare interfaces to add Token-Ring or Ethernet ATM LAN Emulation clients to an existing ATM interface.

- You are adding virtual networks to an existing ESCON Channel Adapter or Parallel Channel Adapter.

To configure a spare interface:

1. Access the CONFIG process by entering **talk 6**.
2. Configure the number of spare interfaces for the device using the **set spare-interfaces** command.
3. Exit the CONFIG process by pressing **Ctrl-P**.
4. Reload the device.

**Example:**

```
* talk 6
Config> set spare 2
Config>
*reload
Are you sure you want to reload the gateway? (Yes or [No]) yes
```

When the device reloads, the spare interfaces are installed as null devices.

To use one of the spare interfaces:

1. Insert the new adapter into the adapter slot.

   **Note:** If you are using an ESCON or Parallel channel adapter, you have the ability to define spare interfaces for each of the attached networks without using additional adapters. In this case, you can bypass this step.
2. Access the CONFIG process by entering **talk 6**.
3. Add an interface or a dial circuit using the **add device** command, if necessary.
4. Configure the spare interface by using the **net** command to configure the interface or add ATM LAN Emulation clients.
5. Configure the various protocols and features using the **protocol** and **feature** commands.
6. Exit the CONFIG process by pressing **Ctrl-P**.
7. Access the GWCON process by entering **talk 5**.
8. Bring the new interface online to the network using the **activate** command.

The following example shows how to configure and activate a new dial circuit on which the IP protocol is enabled. The dial circuit and IP protocol configuration are not shown.

**Example:**

```
*talk 6
Config> add device dial-circuit
Config> net 6
Circuit configuration
Circuit config>

⋮
 Here you would configure the dial circuit

Circuit config> exit
Config> protocol ip
IP>


⋮
 Here you would configure the IP protocol on the dial circuit.

⋮
```

### Using the CONFIG (Talk 6) Process

```
IP> exit
Config>
*talk 5
+activate 6
```

## Restrictions for Spare Interfaces

The **activate** command cannot be used to bring a new interface online to the network under the following circumstances:

- You have already entered a **delete interface** command. The device must be restarted if ***any*** interface has been deleted. You cannot delete a spare interface (indicated by ***null*** in list displays).
- The spare interface is the only interface that enables a protocol or feature. The protocol or feature must already be enabled on an existing interface before it can be used by a spare interface.
- The new spare interface has a header size or trailer size greater than the sizes for other interfaces.
- There is not enough memory to allocate receive buffers for the new interface.

In these cases, you must restart the device to bring the new interface online.

You can configure the following interfaces as spare interfaces, but you cannot bring them online to the network using the **activate** command:

- ATM
- SDLC
- SDLC Relay
- V.25bis
- PPP Multilink master and dedicated link nets

You must restart the device to bring these interfaces online.

You can configure the following protocols on spare interfaces, but you cannot bring them online to the network using the **activate** command:

- LNM
- OSI/DECnet V
- XTP

**Note:** When using the configuration program, use the following to work with spare interfaces:

1. Make the configuration changes for the spare interface on the device
2. Enter the **activate** command on the device to bring the spare interface, protocols, and features online
3. Retrieve the configuration using the configuration program
4. Save the retrieved configuration into the configuration program database

There are also limitations on certain functions. These limitations are:

APPN — To activate this protocol on a spare interface, you must first activate the interface and then configure the protocol on the activated interface.

| | |
|---|---|
| Bandwidth Reservation (BRS) | To configure BRS on a spare interface, you must enable BRS on each network interface where Frame Relay circuits will be active before activating the spare interface. After activating the spare interface, you can then use BRS configuration commands to make changes like adding a traffic class or assigning a protocol to a traffic class. |
| DECnet IV | To activate this protocol on a spare interface, you must first activate the interface and then configure the protocol on the activated interface. Use the DECnet IV **set** command to bring the configuration changes online. |
| Frame Relay | • You cannot activate an FR dial circuit interface unless the dial circuit's base net is already active.<br>• An activate for an FR dial circuit will fail if the frame size, MAC header, or trailer required by the spare interface is larger than other dial circuits already assigned to the base net.<br>• If data compression is not already active in the device, data compression will not work on a spare interface defined for data compression. |
| BGP | Use the BGP **reset neighbor** command to activate new neighbors. |
| IPX | Use the **reset** command to activate static routes, static services, and filter lists on the spare interface. |
| PPP | • If data compression is not already active in the device, data compression will not work on a spare interface defined for data compression.<br>• You cannot activate a spare PPP interface if the device's global buffer is too small to support a 1500-byte PPP MRU.<br>• You cannot activate a PPP dial circuit interface unless the dial circuit's base net is already active.<br>• An activate for a PPP dial circuit will fail if the frame size, MAC header, or trailer required by the spare interface is larger than other dial circuits already assigned to the base net. |
| Bridging | • Bridging was not already active.<br>• NetBIOS filters are defined on the spare interface.<br>• The spare interface caused a change to the bridge personality or behavior (for example, adding SR port to pure TB bridge or SR-TB conversion enabled). |
| IP | Use the reset IP command to bring configuration changes online for access-controls and packet-filters. |
| Channel Virtual Nets (MPC, LCS, LSA) | The spare interface cannot be activated if its configured subchannels are already in use by or shared with another interface. |

WAN Restoral/ WAN Reroute

The spare interface cannot be activated if any of the following conditions are true:

- The spare interface is configured as a WRS primary, and its configured WRS secondary is already a WRS primary or WRR primary or WRR alternate.
- The spare interface is configured as a WRS primary, and its configured WRS secondary is already actively restoring some other WRS primary.
- The spare interface is configured as a WRS secondary, and its configured WRS primary is already a WRS secondary or WRR primary or WRR alternate.
- The spare interface is configured as a WRS secondary, and its configured WRS primary is already actively being restored by some other WRS secondary.
- The spare interface is configured as a WRR primary, and its configured WRR alternate is already a WRS primary or WRS secondary or WRR primary or WRR alternate.
- The spare interface is configured as a WRR alternate, and its configured WRR primary is already a WRS primary or WRS secondary or WRR alternate.
- The spare interface is configured as a WRR alternate, and its configured WRR primary is already actively being rerouted by some other WRR alternate.

## Resetting Interfaces

Occasionally, you might need to change the configuration of a network interface along with its bridging and routing protocols without restarting the device. The **reset** command allows you to disable a network interface and then enable it using new interface, bridging and routing configuration parameters.

The interface, protocols and features configuration parameters are changed using the CONFIG process (talk 6) commands. The talk 6 commands affect the contents of the configuration memory. The configuration changes are activated by issuing the GWCON process (talk 5) **reset** command.

To reset an interface:

1. Access the CONFIG process (talk 6).
2. Use the **net** command and other commands to change configuration parameters.
3. Use the **protocol** and **feature** commands to change the interface-based configuration parameters.
4. Exit the CONFIG process by pressing **Ctrl-P**.
5. Access the GWCON process (talk 5).
6. Use the **reset** command to reset the interface and the protocols and features on the interface.

**Example:**

```
*talk 6
Config>net 1
PPP Config>

. . . change PPP parameters . . .

PPP Config>exit
Config>protocol ipx
```

```
IPX Config>

. . . change IPX parameters on the PPP interface . . .

IPX Config>exit
Config>
*talk 5
+reset 1
Resetting net 1 PPP/0...successful
```

**Note:** When using the configuration program, use the following to make configuration changes to existing interfaces:

1. Make the configuration changes for the interface on the device
2. Enter the **reset** command to reset interface, protocol and feature parameters
3. Retrieve the configuration using the configuration program
4. Save the retrieved configuration into the configuration program database

## Restrictions for Resetting Interfaces

The **reset** command cannot be used to reset a network interface under the following conditions:

- You have already entered a **delete interface** command. The device must be restarted if any interface has been deleted.
- You have changed the hardware or data link type. For example, changing the data link type from PPP to Frame Relay.
- You have configured a larger MTU.
- You have configured a routing protocol or bridging on the interface, but that routing protocol or bridging is not currently active in the device.

In these cases, you must restart the device to bring the configuration changes online.

You can change the configuration parameters of the following types of interfaces, but you cannot bring the configuration changes online using the **reset** command:

- ATM
- PPP Multilink master and dedicated link nets
- ISDN PRI
- X.25
- SDLC
- SDLC Relay
- V.25bis

You must restart the device to bring these configuration changes online.

You can change the configuration parameters of the following protocols and features, but you cannot bring the configuration changes online using the **reset** command:

- AppleTalk
- Vines
- OSI/DECnet V
- LNM
- XTP

- WAN Restoral
- WAN Reroute

You must restart the device to bring these configuration changes online.

There are also limitations on certain functions. These limitations are:

| | |
|---|---|
| PPP dial circuits | A PPP dial circuit cannot be reset if any of the dial circuit parameters have changed. |
| Frame Relay dial circuits | A Frame Relay dial circuit cannot be reset if any of the dial circuit parameters have changed. |
| Compression | Compression requires large header and trailer sizes. Unless compression is already enabled on some other interface, it is likely that the header and trailer sizes will be too small. In this case, compression will be automatically disabled on the interface and an ELS message will be logged (rather than causing the entire reset interface to fail). |
| ESCON base net | Resetting the ESCON base net automatically resets all associated virtual nets. |
| PCA base net | Resetting the PCA base net automatically resets all associated virtual nets. |
| Channel virtual nets (MPC,LCS, LSA) | Must reset the ESCON or PCA base net in order to reset the following virtual net parameters: Subchannels (adding/deleting/changing/moving), LAN type, LAN number, Block timer, Acknowledgement length. |
| Bridging | • Bridging was not already active.<br>• NetBIOS filters are defined on the interface you are resetting.<br>• The reset interface caused a change to the bridge personality or behavior (for example, adding SR port to pure TB bridge or SR-TB conversion enabled). |
| BGP | Use the BGP **reset neighbor** command to bring neighbor configuration changes online. |
| APPN | Use the **activate_new_config** command to bring configuration changes online. |
| IPX | Use the IPX **reset** command to bring configuration changes online for static routes, static services, and filter-lists. |
| DNA IV | Use the DNA IV **set** command to bring configuration changes online. |
| SNMP | Use the SNMP **revert** command to bring configuration changes online. |

# Chapter 9. Configuring the CONFIG Process

This chapter describes the CONFIG process configuration and operational commands. It includes the following sections:

- "Entering and Exiting CONFIG"
- "CONFIG Commands"

## 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 9 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** .)

## 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 6.

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

*Table 6. CONFIG Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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, |
| Enable | Enables login from a remote console, enables modem use, |
| 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. |

*Table 6. CONFIG Command Summary (continued)*

| Command | Function |
|---|---|
| Load | Lists, adds, or deletes optional software packages. |
| Network | Enters the configuration environment of the specified network. |
| Patch | Modifies the router's global configuration. |
| Performance | Provides a snapshot of the main processor utilization statistics. |
| 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, number of spare interfaces, location, and contact person. |
| System | Retrieves dumps |
| 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. |
| Write | Writes the current configuration information to the nonvolatile memory. |

# 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:**

**add** device . . .
isdn-address . . .
ppp-user
tunnel-profile
user . . .
v25-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 Operating Processes" on page 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, "List" on page 84.)

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.

**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 user profile to the local PPP user data base. You need to configure PPP users if you are using PPP authentication protocols, PPP encryption, and want the PPP user data base to be locally stored and managed by the device. If you want PPP user information to be obtained from a RADIUS, TACACS, or TACACS+ server then you should configure the Authentication feature instead of configuring local PPP users.

A user profile stored locally on the device consists of the following:

**User Name**
Name to identify user.

**Password**
A password known to the user and the device. The password can be up to 31characters in length and consist of any alphanumeric character. The password is case sensitive.

**Will this user be tunnelled?**
Specifies whether this dial-in user should be tunneled to an LNS destination. If you answer “yes”, you will be prompted for information about the LNS.

**Hostname to use when connecting to this peer:**
Specifies the local hostname of this LAC that is passed as identification to the LNS during tunnel setup.

**Tunnel Server endpoint:**
Specifies the IP address of the LNS to which this user is tunnelled.

**Type of Route**
Either “Host Route” or “Net Route.”

A host route is generally applied for single-user access. A net route is generally applied to a network access. A net route allows you to enter a net mask.

**User IP Address**
IP address to be assigned to a user.

A user profile-based IP address to offer to a dial-in client if requested. There are a number of ways for a 2216 to obtain an IP address for a dial-in client. See “IP Control Protocol” on page 522 for more information.

**Net-Route Mask**
Mask for a network user.

If the dial-in user is connecting to a DIALs-enabled PPP interface, the router automatically adds a temporary static route to that client for the duration of the PPP session. Typically, this static route has a net mask of 255.255.255.255, which implies that there is a single IP host at the other end of the PPP link. However, the net mask can be overridden. If configured, this mask is used when adding the temporary route. An example of is a small router with a single network of hosts that dials into a DIALs-enabled router. The single route to the small office router will be automatically installed based on the user profile, making it unnecessary to configure routing protocols between the two hosts and cutting down on routing traffic overhead over a potentially slow link.

**Time-Allotted**
The length of time a DIALs user can be connected. This is the total for this session, and should not be confused with an inactivity timer.

**Valid Values:** 0 - 71 827 788 minutes (0=unlimited)

**Default Value:** 0

**Callback type**
Call back method, either “Roaming” or “Required.”

**Encryption**
enable encryption.

You add a PPP user for each remote router or DIALs client that can connect to the device you are configuring.

You are prompted for the PPP user name, password, IP address, and encryption key if encryption should be enabled for the user.

The input parameters are used as follows:

- The PPP user name and password are used during PPP authentication. See “PPP Authentication Protocols” on page 517.
- The encryption key is used by the PPP Encryption Control Protocol (ECP). See “Chapter 70. Overview of Encryption” on page 843.
- The IP address is the address to be assigned to the user.

  A user profile-based IP address to offer to a dial-in client if requested. There are a number of ways for a 2216 to obtain an IP address for a dial-in client. See “IP Control Protocol” on page 522 for more information.

- The net mask is entered when a dial-in user is a network type. The mask defaults to 255.255.255.255 for a single user.

  If the dial-in user is connecting to a DIALs-enabled PPP interface, the router automatically adds a temporary static route to that client for the duration of the PPP session. Typically, this static route has a net mask of 255.255.255.255, which implies that there is a single IP host at the other end of the PPP link. However, the net mask can be overridden. If configured, this mask is used when adding the temporary route. An example of is a small router with a single network of hosts that dials into a DIALs-enabled router. The single route to the small office router will be automatically installed based on the user profile, making it unnecessary to configure routing protocols between the two hosts and cutting down on routing traffic overhead over a potentially slow link.
- The time allotted is used to restrict the amount of time that a PPP user can stay connected.
- The call back parameters are used to specify whether the router will call back the user and what number to call back. See “Configuring PPP Callback” on page 520 for additional information.

You can add up to 500 PPP users.

**Example:**Adding a PPP dials user with a hostroute

```
Enter name:  []? dialshost
Password:
Will 'dialshost' be tunneled? (Yes, No): [No]
Is this a 'DIALs' user? (Yes, No): [Yes]
Type of route?  (hostroute, netroute): [hostroute]
Number of days before password expiry[0-1000] [0]?
IP address: [0.0.0.0]?
Enter hostname for dynamic DNS:  []?
Give 'dialshost' default time allotted ? (Yes, No): [Yes]
Enable callback for 'dialshost' ? (Yes, No): [No]
Will 'dialshost' be able to dial-out ? (Yes, No): [No]
Enable encryption for this user/port (y/n) [No]:
Disable 'dialshost' ? (Yes, No): [No]
```

**Example:**Adding a PPP dials with a netroute

```
Enter name:  []? dialsnet
Password:
Enter again to verify:
Will 'dialsnet' be tunneled? (Yes, No): [No]
Is this a 'DIALs' user? (Yes, No): [Yes]
Type of route?  (hostroute, netroute): [hostroute] n
Number of days before password expiry[0-1000] [0]?
IP address: [0.0.0.0]?
Net mask: [0.0.0.0]?
Enter hostname for dynamic DNS:  []?
Give 'dialsnet' default time allotted ? (Yes, No): [Yes]
Enable callback for 'dialsnet' ? (Yes, No): [No]
Will 'dialsnet' be able to dial-out ? (Yes, No): [No]
Enable encryption for this user/port (y/n)  No]:
Disable 'dialsnet' ? (Yes, No): [No]
```

**Example:**Adding PPP no dials

```
Enter name:  []? nodialsnet
Password:
Enter again to verify:
Will 'nodialsnet' be tunneled? (Yes, No): [No]
Is this a 'DIALs' user? (Yes, No): [Yes] n
Number of days before password expiry[0-1000] [0]?
IP address: [0.0.0.0]?
Enable encryption for this user/port (y/n) [No]:
Disable 'nodialsnet' ? (Yes, No): [No]
```

**Example:**Adding a PPP tunneled user

```
Enter name:  []? tunneluser
Will 'tunneluser' be tunneled? (Yes, No): [No] y
Enter hostname to use when connecting to this peer: []?
Tunnel-Server endpoint address: [0.0.0.0]?
```

**tunnel** *tunnel_name*

Gives a tunnel peer access through an IP network to the router. This peer is then authorized to initiate tunneled PPP sessions into the router. To configure a tunnel you must specify:

**Name** The hostname of the tunnel peer.

**Hostname to use when connecting to this peer**
The local hostname to use when connecting to this peer. This name is used for identification of the host on the peer.

**Shared Secret**
The secret shared between the LAC and LNS. It must be exactly the same on both ends of the tunnel.

**Tunnel-Server endpoint**
The IP address of the tunnel peer (LAC or LNS).

**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 permission levels defined in Table 7.

When users are added, set login authentication to local. Otherwise a remote server must be used.

*Table 7. Access Permission*

| Permission Level | 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 permission level for the user: A, O, or M (see Table 7 on page 72).

**v25-bis-address**
Adds the local and remote numbers of the V.25bis 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 49. Using the V.25bis Network Interface" on page 613.

**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 6. Using BOOT Config to Perform Change Management" on page 41.

**Syntax:**

**boot**

## Change

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

**Syntax:**

change — device . . .
password
ppp_user . . .
tunnel-profile

**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 you select the “swap” option, the source and target slots are swapped in all the interface records in which they appear.
2. If you select 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 75.

**Example - Change (replace) slot 1 on interface 1 to occupied slot 2:**

```
Config>list dev
Ifc 0 CHARM ATM                      Slot: 2  Port: 1
Ifc 1 CHARM ATM                      Slot: 1  Port: 1

Config>change device
Which configured slot would you like to change? (1, 2) [2]? 1
Which slot would you like to change to? (1-2)  [1]? 2

Configuration for slot 2 already exists. You can:
a - abort this operation
r - replace configuration
    (Interface configuration for slot 1 will become interface
     configuration for slot 2.  Interface records for slot 2
     will be deleted!)
s - swap configuration (slot 1 will be swapped with slot 2.)
r

Moved slot 2 to slot 1 in 1 intf (port) record...

Config>list dev
Ifc 0 CHARM ATM                      Slot: 2  Port: 1
```

**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 the note on 2 on page 74 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 - Slot 1 on interface 1 replaces slot 6, interface 1 is deleted:

```
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.)
r

Moved slot 6 to slot 1 in 1 intf (port) record...

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: 2  Port: 0
Ifc 3 8 port EIA-232E/V.24 PPP      Slot: 2  Port: 1
Ifc 4 8 port EIA-232E/V.24 PPP      Slot: 2  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
```

The slot 6 record replaces the original slot 1 record and the other interfaces are renumbered.

### Example - Change slot 2 to unoccupied slot 5:

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

Changed slot 2 to slot 5 in 3 intf (port) records...

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: 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
```

Interfaces 2, 3 and 4 that were previously configured in slot 2 are now configured in slot 5.

### 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** encryption-key
parameters
password

**encryption-key**
Changes the encryption key for a PPP user. The following example shows the dialog for changing an encryption key.

**Example - Change Encryption key:**

```
Config>change ppp_user encryption-key
Enter user name:  []? leslie
Enable encryption for this user/port (y/n) [No]:y
Encryption key should be 16 characters long.
Encryption Key (16 characters ) in Hex(0-9, a-f, A-F):
Encryption Key again (16 characters) in Hex(0-9, a-f, A-F):
User 'leslie' has been updated
Config>
```

**parameters**

Changes all of the ppp-user options for a user. This parameter works similar to the **add ppp_user** except that the values shown within the [] are the current values and the change command does not verify the changes or list them back to you when you are done. See “Add” on page 68 for details about the **add ppp_user** command.

**password**

Changes the password for the PPP user.

**Example - Change password:**

```
Config>change ppp_user password
Enter user name:  []? sam
Password:
Enter password again:
User 'sam' has been updated
Config>
```

**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])
```

**tunnel-profile**

Changes the configuration for a tunnel peer.

```
Config>change tunnel-profile
Enter name:  []? lac.org
Enter hostname to use when connecting to this peer: [lns.org]?
set shared secret? (Yes, No): [No]
Tunnel-Server endpoint address: [11.0.0.1]? 11.0.0.2

profile 'lac.org' has been updated
Config>
```

## Clear

Use the **clear** command to delete the router’s configuration information from nonvolatile configuration memory.

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

**Syntax:**

**clear**

all

ap2 (AppleTalk 2)

arp (ARP)

asrt (Adaptive Source Route Protocol)

appn (Advanced Peer-to-Peer Networking)

atm (Asynchronous Transfer Mode)
auth (Authentication)
bgp (Border Gateway Protocol)
boot
brs (Bandwidth Reservation)
cmprs (Data Compression)
dls (Data Link Switching)
device
dialer-circuit
dn (DECnet)
dvmrp (Distance Vector Multicast Routing Protocol)
els (Event Logging System Information)
fr (Frame Relay)
hdlc
ip (IP)
ipx (Novell IPX)
isdn
osi (OSI)
ospf (OSPF routing protocol)
ppp (Point-to-Point)
sdlc
snmp
srb (Source Route Bridge)
srly (SDLC Relay)
stb (Spanning Tree Bridge)
tcp/ip-host
time (Time of day information)
user
v25bis
vines (Banyan VINES)
wrs (WAN Restoral feature)
x25

To clear a process from nonvolatile 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. 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.

**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 or range of interfaces 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:**

**delete** interface . . .
isdn-address
ppp_user . . .
tunnel
user . . .
v25-bis-address

**interface [***intfc#* **or** *intfc#range***]**

To delete an interface, enter the interface or network number as part of the command. (Only devices that were added with the **add device** command 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.

To delete a range of interfaces, specify the first and last interface in the range separated by a hyphen, as shown in the following example:

```
delete interface 13-21
```

You can also enter an interface number or range of interface numbers, when prompted.

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

**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** *user_name*
Deletes a user from the PPP user data base.

**tunnel-profile**
Deletes a tunnel from the tunnel profile database.

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

**v25-bis-address** *address-name*
Removes a previously added V25bis address.

**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

You can also use the disable command to disable an interface, memory dumping, or rebooting when a serious error occurs.

**Syntax:**

**disable** console-login
interface . . .
dump-memory . . .
reboot-system . . .

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

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

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

**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
```

## Enable

Use the **enable** command to allow login from a remote console,

**Syntax:**

enable

- console-login
- interface . . .
- dump-memory . . .
- reboot-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.
```

**Attention:** Before enabling console login, save the configuration with console login disabled. If login authentication is set to a remote server using Radius or Tacacs+ and the router is unable to reach the authentication server, then access to the router is denied. By disabling the console login, a lock-out situation is prevented.

**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
```

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

## 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 12. Using the Event Logging System (ELS)” on page 117 for information about ELS.

**Syntax:**

**event**

# Feature

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

**Syntax:**

**feature** [*feature#* or *feature-short-name*]

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.

To access a feature's configuration prompt, enter the **feature** command followed by the feature number or short name. Table 8 lists available feature numbers and names.

*Table 8. 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 |
| 6 | QOS | Quality of Service |
| 7 | CMPRS | Data Compression |
| 8 | NDR | Network Dispatcher |
| 10 | AUTH | Authentication |
| 11 | IPSec | IP Security feature user configuration |
| 12 | LAYER | Layer 2 Tunneling Protocol |
| 13 | NAT | Network Address Translator user configuration |

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.

# List

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

**Syntax:**

**list** configuration

devices

isdn-address

patches . . .

ppp_users . . .

tunnel-profile

users . . .

v25-bis-address

vpd

**devices [***device* **or** *devicerange***]**

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.

You can also specify a range of devices to list as shown in the following example:

```
list dev 2-5
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
```

**Note:** If you do not specify an interface number or a range of interfaces, all interfaces are displayed.

**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
```

**Note:** The number of receive buffers noted are exceptions from the receive buffer defaults. The **set receive buffers** command is discussed under “Set” on page 92.

**configuration**

Displays configuration information about the router.

**Example: `list configuration`**

```
Hostname: [none]
Maximum packet size: [autoconfigured]
Maximum number of global buffers: [autoconfigured]
Number of spare interfaces: 0
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
28  APPN  Advanced Peer-to-Peer Networking [HPR]
29  NHRP  Next Hop Routing Protocol
30  APPN  Advanced Peer-to-Peer Networking [ISR]

Configurable Features:
Num Name  Feature
0   WRS   WAN Restoral
1   BRS   Bandwidth Reservation
2   MCF   MAC Filtering
6   QOS   Quality of Service
7   CMPRS Data Compression Subsystem
8   NDR   Network Dispatching Router
10  AUTH  Authentication

26176 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 specific PPP user profile parameters.

**Example:** List of PPP users when DIALs is not in the software load

```
Config> list ppp_users
List (Name, Verb, User, Addr, Encr):

       PPP User Name: joe
     User IP Address: Interface Default
          Encryption: Not Enabled
```

**Example:** List of PPP users when DIALS is in the software load

```
Config> list ppp_users
List (Name, Verb, User, Addr, Call, Time, Dial, Encr):

       PPP User Name: joe
     User IP Address: Interface Default
      Net-Route Mask: 255.255.255.255
            Hostname: <undefined>
       Time-Allotted: Box Default
      Call-Back Type: Not Enabled
            Dial-Out: Not Enabled
          Encryption: Not Enabled
```

When you enter **list ppp_users**, the software will prompt you to enter one of the following:

**Name** List all of the names in the database.

**Verb** List verbose information about each user. List all information pertaining to each user profile.

**User** List verbose information about a single user.

**Addr (address)**
List IP address information for each user, including IP Address, net mask and hostname.

**Call (callback)**
List callback information for each user, including the type of callback and number.

**Time** List time allowed configured for each user.

**Encr (encryption)**
List whether encryption is enabled for each user.

**tunnel-profile**
Displays the tunnel-profile parameters.

**Example:**

```
Config>list tunnel-profile
Tunnel Name     Server Endpoint  Type      Medium    Local Hostname
lac.org         11.0.0.1         L2TP      IP        lns.org
lns-1           11.0.0.170       L2TP      IP        lac-1

2 records displayed.

Config>
```

**Tunnel Name**
Specifies the configured name for the peer.

**Server Endpoint**
The IP address of the peer.

**Type** Specifies the type of peer connection.

**Medium**
Specifies the protocol that the tunnel is using.

**Local Host Name**
Specifies the name configured for use when connecting to the peer.

**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 V25bis address configurations. The V25bis 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 49. Using the V.25bis Network Interface” on page 613.

```
Example:
list v25-bis-address
Address assigned name            Network Address
---------------------            ---------------
```

```
v25-1                          8982800
v25-2                          8980001
westboro                       1-666-555-4444
```

**vpd** Displays the hardware and software vital product data.

# Load

Use the **load** command to list packages in the software load that are available but not configured, or packages that are configured in the software load. The **load** command is also used to add or delete a sofware package.

**Syntax:**

**load** add package *packagename*

delete package *packagename*

list . . .

The software is divided into multiple load modules. These load modules are grouped into software packages. Some of these software packages are optional because, although they are shipped with the product, they are not automatically loaded.

Software packages containing encryption are available from the 2216 Web server accessable using the Internet.

To load and run optional software packages:

1. Add the package using the **load add** command.
2. Reboot. This action loads the optional software into the device's memory.
3. Configure the optional software.
4. Save the configuration.
5. Reboot the device. This action enables the software with the new configuration.

**add package** *packagename*
Adds a software package to the software. The *packagename* is the name of the package of load modules you want to include in the software.

**Example: load add package appn**

**delete package** *packagename*
Removes a software package to the software. The *packagename* is the name of the package of load modules you want to remove from the software.

**Example: load delete package appn**

**list** Lists either the packages in the software load that are available but not configured, or the packages that are configured in the software load. You can specify one of the following:

**available**
Lists the software packages in the current software load that are not configured.

**configured**
Lists the software packages in the current software load that are configured.

## 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.

**Syntax:**

**network** *interface#*

**Notes:**

1. 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 (*).
2. 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
- ip-default-ttl
- ip-mtu
- lnm-link-via-tbport
- more-lines
- mosheap-lowmark
- ospf-import-rate
- ping-size
- ping-ttl
- ppp-echo
- relax-jate
- rip-static-suppress

**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.

**dls-ignore-lfs** *new value*
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** *new value*
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.

**ip-default-ttl** *#_of_packets*
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 3 Release 1* .) This patch variable remains for compatibility with configurations from older releases.

**ip-mtu** *bytes*
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.

**lnm-link-via-tbport** *new value*
Allows LNM to link to a token-ring over an Ethernet transparent bridge (TB) port.

When set to 1, the LNM link is allowed.

When set to 0, the default, the LNM link is not allowed.

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

**mosheap-lowmark** *new value*
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** *rate*
Number of routes imported per second.

**ping-size** *bytes*
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 3 Release 1* .)

**ping-ttl** *seconds*
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 3 Release 1*.

**ppp-echo** *new value*
When set to 1, the device will not send PPP Echo Requests on any PPP interface. PPP Echo Requests are sent to remote devices as part of PPP maintenance to ensure the remote device is operational. Consider enabling this variable when running PPP on a slow line and using that line to transmit large data packets such that the PPP maintenance packets are not exchanged often enough to keep the PPP interface up.

**relax-jate**
Relaxes JATE ISDN restriction.

**rip-static-suppress** *new value*
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.

## Performance

Use the **performance** command at the GWCON> prompt (+) to enter the configuration environment for performance. See "Chapter 14. Configuring and Monitoring Performance" on page 181 for more information.

## Protocol

Use the **protocol** command at the Config> prompt to enter the configuration environment for the protocol software installed in the router.

**Syntax:**

**protocol** [*prot#* or *prot_name*]

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. 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 nonvolatile memory and are recalled when you restart the router.

## Qconfig

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

**Syntax:**

**qconfig**

**Note:** For complete information on using the Quick Config software provided with your router, see “Appendix A. Quick Configuration Reference” on page 909.

## 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 . . .
- logging level
- packet-size
- prompt-level
- receive-buffers
- restart-count
- spare-interfaces

**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.

**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, SDLC, 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.

**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.

**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.

**hostname** *name*

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.

**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 affects 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.

**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 to 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.

**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.

**logging level** *#*

Controls the output of messages that have not yet been converted to the ELS. (Refer to for more information about the ELS.) The logging level is recorded in the configuration. When the router is powered on or restarted, the logging level takes effect and determines message output. The default logging level is 76. Logging level 0 equates to no logging level.

```
Example: set logging level 76
```

**packet-size** *max_packet_size_in_bytes*

Establishes or changes the maximum size for global buffers and receive buffers. If you specify a value of 0 as the maximum packet size, the size of

receive buffers for an interface is based on that interface's configured packet size and the packet size of global buffers are autoconfigured. If you specify a non-zero value, the configured value is used as the global buffer packet size and any interfaces that have a configured packet size that is larger than the maximum packet size will use the maximum packet size for their receive buffers. A value of 0 (for autoconfigure) is the default.

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

**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 9.

**Example:**

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

*Table 9. 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:<br>`Config> set prompt`<br>`What is the new MOS prompt [y]? $n`<br>`hostname:: Config>` |
| $t | Displays the time. For example:<br>`Config> set prompt.`<br>`What is the new MOS prompt [y]? $t`<br>`02:51:08[GMT-300] Config>` |
| $d | Displays the current date-month-year. For example:<br>`Config> set prompt.`<br>`What is the new MOS prompt [y]? $d`<br>`26-Feb-1997 Config>` |
| $v | Displays the software VPD information in the following format:<br>`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 $. |

*Table 9. Additional Functions Provided by the Set Prompt Level Command (continued)*

| Special Characters | Function Provided by the Set Prompt Level Command |
|---|---|
| **Note:** You can combine these commands. For example:<br>`Config> set prompt`<br>`What is the new MOS prompt [y]? $n::$d`<br>`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 and 10/100 Mbps Ethernet interfaces. For ISDN PRI, the number of receive buffers is fixed at 5 per B-channel, 115 for T1 and 150 for E1.

(On some devices, the maximum value is restricted further, as shown in 10.) 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.

*Table 10. Default and Maximum Settings for Interfaces*

| Interface | Default | Maximum |
|---|---|---|
| ATM | 80 | 80 |
| ETH | 40 | 100 |
| Serial | 24 | 60 |
| TKR | 40 | 100 |
| FDDI | 80 | 80 |
| HSSI | 60 | 255 |

**spare-interfaces** ***n***

Defines ***n***, the number of spare interfaces, for this device. See “Configuring Spare Interfaces” on page 60 for additional information.

## System Memory Dump

Use the **system memory dump** command to retrieve the memory image file from the installed hard disk after a serious error has occurred.

**Syntax:**

**system** retrieve *address filepath*

**retrieve**

Uses TFTP to send the memory image to a remote location, with a destination TFTP file address, path, and file name supplied by the user.

If memory dumping is disabled, the function is aborted and the following message is displayed:

```
Image file transfer aborted:  function disabled
```

If the memory file is not present on the hard disk, or if the hard disk has been removed, the function is aborted and the following message is displayed.

```
Image file transfer aborted:  image file not found
```

```
Example: system retrieve
Destination IP address (0.0.0.0) 2.2.2.2
Fully qualified destination path/file name (tmp/dump.dat)
The memory image file is nnnnn bytes long.
Proceed? (Y/[N])
```

## 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 reinitialization.

**Syntax:**

**time** host . . .
list
offset
set . . .
sync . . .

**host** *IP_address*
Sets the IP address of the RFC 868-compliant host that 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.

**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**.

**Valid values:** -720 to 720

**Default value:** 0

**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.

## 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 89.

**Syntax:**

**unpatch** *variable_name*

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

## Update

Use the **update** command to update the configuration memory when you receive a new software load.

**Syntax:**

**update** version-of-SRAM

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.

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

## Write

Use the **write** command to save a configuration to the device before reloading.

**Syntax:**

**write**

If you fail to issue the write command and try to reload the device, you will be asked if you want to save the configuration. The configuration is saved in the next CONFIG on the hard disk in the bank you are currently using.

# Chapter 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands

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

- "What is GWCON?"
- "Entering and Exiting GWCON"
- "GWCON Commands"

## 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.

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 enter the **talk 5**

```
* 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 11 on page 100.

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

*Table 11. GWCON Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Activate | Enables a newly configured spare interface. |
| 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 or slot off line. |
| Enable | Enables all interfaces of an adapter. |
| 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. |
| Network | Enters the console environment of the specified network. |
| Performance | Provides a snapshot of the main processor utilization statistics. |
| Protocol | Enters the command environment of the specified protocol. |
| Queue | Displays buffer statistics for a specified interface. |
| Reset | Disables the specified interface and then re-enables it using new interface, protocol and feature configuration parameters. |
| Statistics | Displays statistics for a specified interface. |
| Test | Enables a disabled interface or tests the specified interface. |
| Uptime | Displays time statistics for the router. |

## Activate

Use the **activate** command to enable a spare interface on this device. See “Configuring Spare Interfaces” on page 60 for more information.

**Syntax:**

**activate** *interface#*

## Buffer

Use the **buffer** command to display information about packet buffers assigned to each interface or range of interfaces.

**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:**

**buffer** [*network#* or *range_of_network#*]

To display information about multiple interfaces, specify the range_of_network# (or a combination of *network#* and *range_of_network#*). For example, specifying **buffer 0 3 25-50** displays the information for nets 0, 3, and 25 through 50.

**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:**

**clear** *interface#*or *range_of_interface#*

To clear information about multiple interfaces, specify the range_of_network# (or a combination of *interface#* and *range_of_interface#*). For example, specifying **clear**

**0 3 25-50** clears the information for nets 0, 3, and 25 through 50.

# 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:**

**configuration**

To display information about multiple interfaces, specify the range_of_network# (or a combination of *network#* and *range_of_network#*). For example, specifying **configuration 0 3 25-50** displays the information for nets 0, 3, and 25 through 50.

**Example:**

```
configuration

Multiprotocol Access Services


5765-C90 Feature 2802 V1 R2.0 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 gives the product name.
- The second line lists the program/product number, Feature Number, Version, Release, PTF and RPQ information.
- The remaining lines list the configured protocols, followed by the configured features.

The following information is displayed for protocols:

**Num** Number that is associated with the protocol.

**Name** Abbreviated name of the protocol.

**Protocol**
Full name of the protocol.

The following information is displayed for features:

**Num** Number associated with the feature.

**Name** Abbreviated name of the feature.

**Feature**
Full name of the feature.

The following information is displayed for 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 correctly. 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 the router is reinitialized, 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 reinitialized.
- 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 reinitialized.

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 63. The WAN Reroute Feature” on page 759 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.

Not Present is also used as the state for a null device. Spare interfaces are displayed as null devices until they are activated.

**HW Mismatch**
Indicates that the configured adapter type does not match the adapter type that is actually present in the slot.

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

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

**Note:** If the interface you are disabling is configured as an alternate WAN Reroute interface, you are asked if you want to disable any WAN Reroute primary/alternate pairings that include this alternate interface. If you answer *yes*, the interface is disabled and is no longer available to backup a primary interface. If you answer *no*, the alternate interface is disabled but WAN Reroute will attempt to bring it up if its corresponding primary interface goes down. You want to disable WAN Reroute on an alternate interface if you are disabling the interface so that you can remove its adapter. See “Chapter 63. The WAN Reroute Feature” on page 759, “Chapter 61. Using WAN Restoral” on page 739 , and “Chapter 62. Configuring and Monitoring WAN Restoral” on page 743 for additional information.

**Syntax:**

| | |
|---|---|
| **disable** | interface *interface#* |
| | slot *slot#* |

## Enable

Use the **Enable** command to enable all interfaces of an adapter. This performs the same action as the **test** command (See “Test” on page 112) but performs the action for each interface using the adapter in the specified slot.

**Syntax:**

| | |
|---|---|
| **enable** | slot *slot#* |

## Error

Use the **error** command to display error statistics for the network. This command provides a group of error counters.

**Syntax:**

| | |
|---|---|
| **error** | [*network#* or *range_of_network#*] |

To display information about multiple interfaces, specify the range_of_network# (or a combination of *network#* and *range_of_network#*). For example, specifying **error 0 3 25-50** displays the information for nets 0, 3, and 25 through 50.

**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 12. Using the Event Logging System (ELS)" on page 117 for information about the Event Logging System and its commands. Use the **exit** command to return to the GWCON process.

**Syntax:**

**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.

To access that feature's console prompt, enter the **feature** command at the GWCON prompt followed by the feature number or short name. Table 8 on page 84 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:**

**feature** *feature#* or *feature-short-name*

## Interface

Use the **interface** command to display statistical information about the network interfaces (for example, FDDI,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 about 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:**

**interface** [*interface#* or *range_of_interface#*]

To display information about multiple interfaces, specify the range_of_network# (or a combination of *interface#* and *range_of_interface#*). For example, specifying **`interface 0 3 25-50`** displays the information for nets 0, 3, and 25 through 50.

**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:**

**memory**

**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”.)

**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.

**Syntax:**

**network** *interface#*

At the GWCON prompt (+), enter the **configuration** command to see the protocols and networks for which the router is configured. See “Configuration” on page 102 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 or change. 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 operating commands.

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.25bis
- X.25
- ATM
- ISDN
- Dial-In
- Multilink PPP (MP)

## Performance

Use the **performance** command at the Config> prompt to enter the monitoring environment for performance. See “Chapter 14. Configuring and Monitoring Performance” on page 181 for more information.

## 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.

**Syntax:**

**protocol** *prot#*

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 “Accessing the Configuration Process, CONFIG (Talk 6)” on page 14 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.

## 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.

**Syntax:**

**queue** *interface#*or *range_of_interface#*

To display information about multiple interfaces, specify the range_of_network# (or a combination of *interface#* and *range_of_interface#*). For example, specifying **queue 0 3 25-50** displays the information for nets 0, 3, and 25 through 50.

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.

**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.

## Reset

Use the **reset** command to disable the specified interface and then re-enable it using new interface, protocol and feature configuration parameters. See “Resetting Interfaces” on page 64 for more information.

**Syntax:**

**reset** *interface#*

## Statistics

Use the **statistics** command to display statistical information about the network software, such as the configuration of the networks in the router.

**Syntax:**

**statistics** *interface#*or *range_of_interface#*

To display information about multiple interfaces, specify the range_of_network# (or a combination of *interface#* and *range_of_interface#*). For example, specifying

**statistics 0 3 25-50** displays the information for nets 0, 3, and 25 through 50.

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.

**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 the **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.

**Syntax:**

**test** *interface#*

**Note:** For this command to work, you must enter the ***complete*** name of the command followed by the interface number.

Enter the interface or network number as part of the command. To obtain the interface number, use the GWCON **configuration** command. For example, when testing starts, the console displays the following message:

```
Testing net 0 TKR/0...
```

When testing completes or fails, or when 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
```

Some interfaces may take more than 30 seconds before testing is done.

**Note:** If the interface you are testing is configured as an alternate WAN Reroute interface, you are prompted:

- If you want to enable the interface's primary-alternate pairings if WAN Reroute is currently disabled for the alternate interface.

  If you answer *yes*, the same action occurs as when you enter the **t 5 enable alternate-circuit** WAN reroute command described in "Chapter 62. Configuring and Monitoring WAN Restoral" on page 743.

- If you want to test the interface.

  Normally an alternate WAN Reroute interface is disabled until it is needed to back up its corresponding primary interface. If you answer *yes*, a self-test is started for the interface. If you answer *no*, a self-test does not occur.

See "Chapter 63. The WAN Reroute Feature" on page 759, "Chapter 61. Using WAN Restoral" on page 739, and "Chapter 62. Configuring and Monitoring WAN Restoral" on page 743 for additional information.

## Uptime

Use the **uptime** command to display time statistics about the router, including the following:

- Number of restarts.
- Number of known crashes.
- Whether the router was last reloaded or restarted.
- Time elapsed since the last reload.
- Time elapsed since the last restart.

**Syntax:**

**uptime**

## GWCON Process

# Chapter 11. The Messaging (MONITR - Talk 2) Process

This chapter explains how to collect and display messages. (Refer to “Chapter 12. Using the Event Logging System (ELS)” on page 117 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 Messaging (MONITR)?”
- “Commands Affecting Messaging”
- “Entering and Exiting the Messaging (MONITR) Process”
- “Receiving Messages”

## What is Messaging (MONITR)?

The MONITR process provides a view of activity inside the router and the networks. MONITR also displays logging messages from the software.

## Commands Affecting Messaging

The following commands affect the messaging process:

- OPCON commands:
  - **divert** temporarily diverts output to a different device.
  - **flush** causes the software to discard the messages it collects.
  - **halt** reverses the action of the divert command.
  - **talk** displays message output.
- CONFIG **set logging disposition** command sets the initial device to which the software sends its output.

## Entering and Exiting the Messaging (MONITR) Process

To enter the messaging process from OPCON enter the **talk 2** command.

The console displays the messages the software has accumulated.

To exit messaging and return to OPCON, enter the OPCON intercept character (the default is **Ctrl-P**).

## Receiving Messages

To receive messages at your console, enter the messaging process as described in the previous section. The software then displays all the messages it has recorded since it was last invoked. While you are connected to the messaging process, it displays all messages as they arrive.

Use the OPCON **divert** and **halt** commands to view software 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 12. Using the Event Logging System (ELS)

This chapter describes the Event Logging System (ELS). The ELS continually logs all events, filtering them according to parameters that you select. A combination of operational 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?

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. Use ELS commands to set up a configuration that sorts out only those messages you feel are important. You can then display the messages on the console terminal screen, log them to a remote workstation,or send the messages to a network management station using Simple Network Management Protocol (SNMP) traps.

The ELS system and the operational 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 the router reinitializes.

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 operating and monitoring environment is used to:

- Temporarily change the default ELS display settings
  - Changes made in the ELS console environment take effect immediately
  - Changes made in the operating/monitoring environment are not stored in nonvolatile 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.

## 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 you restart the router. These commands are described in greater detail later in this chapter.

### Using ELS

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.

To enter the ELS configuration environment from OPCON:

1. Enter the **talk 6** command. The console displays the CONFIG prompt (`Config>`). If the prompt does not appear when you first enter CONFIG, press **Return**.
2. 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.

## 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 5 on page 119 shows the message contents.

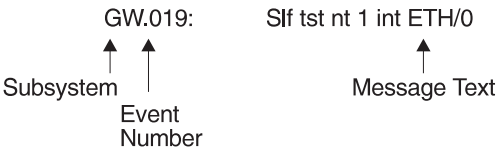



*Figure 5. Message Generated by an Event*

The information illustrated in Figure 5 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 5, **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 (except the events with 'debug' logging level) 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 5, **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. Use the **list subsystem** ELS console command to display the setting of the logging level. Table 12 lists the logging levels and types.

*Table 12. 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 |

*Table 12. Logging Levels (continued)*

| Logging Level | Type |
|---|---|
| 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 |
| DEBUG | Message for debugging |
| ALL | Includes all logging levels |

In Table 12 on page 119, 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**
- **Remote subsystem**
- **Noremote 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 5 on page 119, `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 13 describes the error or packet completion codes. Packet completion codes indicate the disposition of the packets that arrive at the router.

*Table 13. 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 |

*Table 13. Packet Completion Codes (Error Codes) (continued)*

| Code | Meaning |
|---|---|
| 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, 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, 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.

## Using ELS

To use ELS effectively, take the following steps:

- 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.

### Using ELS

When initially viewing 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. When this message appears, 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

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, set up ELS for the messages you want to capture using the ELS console commands in "Chapter 13. Configuring and Monitoring the Event Logging System (ELS)" on page 135.

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.

You can also use remote logging instead of capturing ELS output on a UNIX Host. For more information about remote logging, see "Using and Configuring ELS Remote Logging" on page 125.

## 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.

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, when 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

SRB 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 srb all

   ELS config> exit

   config> Ctrl-P
   ```
4. Restart the routing subsystem. When the subsystem has restarted, enter the following:

   ```
   * t 2
   ```

## ELS Example 3

Router cannot communicate with an IPX server on an Ethernet.

1. 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**.
2. At the GWCON prompt (+), enter **IPX** to access the IPX console prompt (IPX>).
3. 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.)
4. Check the IPX configuration.
5. 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.001. This indicates that the server has a bad Ethernet type field.

## Using and Configuring ELS Remote Logging

The remotely-logged ELS message contains all of the information that is contained in ELS messages found in the monitor queue, as viewed under talk 2, and also contains additional information as shown in Figure 6.

```
Date/Time         IP address      Sequence Number      Local Name      ELS Subsystem Name, &
                  assigned        used for detecting   assigned        Formatted message
                  by the user     missing messages     by the user

Nov 20 12:13:47   5.1.1.1         Msg [0444] from      ** IBM/2216 **  :els: ARP.011 Del ent ...
```

*Figure 6. Syslog Message Description*

Note the following differences in the remote log display:

- The month and day of month in addition to the time, which is always displayed as the time-of-day.
- An IP address, which is the user-specified source IP address. If a DNS server resolves the source IP address to a hostname, then the hostname will be displayed instead of the IP address.
- A Sequence number is added to the message by the source device to assist in detecting dropped messages. See “Remote Logging Output” on page 129 for an explanation of dropped messages. When the sequence number of the message reaches 9999, the next sequence number is 0001.
- A “Local Name” for the source router, to assist in distinguishing between messages from multiple sources. If you do not configure a local name, this field is blank.

## Syslog Facility and Level

Remotely-logged ELS messages are transmitted over the network in UDP packets with the destination port number in the UDP header always equal to 514, the syslog port. To receive and process the UDP packets, the *syslog daemon* (syslogd) must be running in the remote workstation that is receiving and logging the ELS messages. See “Remote Workstation Configuration” on page 126 for details.

Although it is not displayed in the remotely-logged ELS message, every ELS message sent on the network in a UDP packet must be assigned a *syslog_facility* and a *syslog_level*. The syslog daemon uses the combination of facility and level to determine where to route the message. Typically, you want the ELS messages to be written to one or more files in the remote host. Other options include displaying the message on the console, sending the message to one or more users, or sending the message to another workstation.

The commands you use to specify the *syslog_facility* and *syslog_level* values, along with other remote-logging related console commands, are described in “ELS Monitoring Commands” on page 156 and “ELS Configuration Commands” on page 135 . Review these commands before reading through the next section.

## Remote Workstation Configuration

The following configuration assumes that a single 2216 is remote-logging to a single remote workstation. You can configure multiple 2216s to remote-log to the same remote workstation. However, a particular 2216 can log to one and only one remote workstation. The operating system used in this example is AIX 4.2. Your environment may be slightly different. For more information on syslog, refer to the documentation for your operating system.

To perform the configuration on an AIX workstation, you must log in as **root**. To configure the workstation:

1. Create or edit a syslog.conf file to specify where ELS messages with particular *syslog_facility* and *syslog_level* values are to be written. See the bottom of Figure 7 on page 127 for an example of how to specify the message destination. Note that the full pathname of the log files must be specified. The default location for the syslog configuration file is /etc/syslog.conf.
2. Create the files for logging syslog messages that you specified in the syslog.conf file.
3. Start the syslog daemon by entering **syslogd**. To start the syslog daemon from SRC (System Resource Controller), enter **startsrc -s syslogd**. If the pathname of the configuration file is not /etc/syslog.conf, then enter **syslogd -f** *pathname*. To start the syslog daemon in debug mode, enter **syslogd -d**.

   **Note:** Running multiple instances of the syslog daemon is not supported.
4. If the syslog daemon is already running when you create or modify the syslog.conf file, it must be restarted so that the daemon reinitializes the configuration from syslog.conf.
5. Verify the setup by using the **logger** command as follows:

   ```
   logger -p user.alert  THIS IS A TEST MESSAGE (user.alert)
   logger -p news.info  THIS IS A TEST MESSAGE (news.info)
   ```

   If the setup is correct, `THIS IS A TEST MESSAGE...` will be written to the files specified in syslog.conf.

```
# @(#)34       1.9 src/bos/etc/syslog/syslog.conf, cmdnet, bos411, 9428A410j 6/13/93 14:52:39
#
# COMPONENT_NAME: (CMDNET) Network commands.
#
# FUNCTIONS:
#
# ORIGINS: 27
#
# (C) COPYRIGHT International Business Machines Corp. 1988, 1989
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# /etc/syslog.conf - control output of syslogd
#
# Each line must consist of two parts:-
#
# 1) A selector to determine the message priorities to which the
#    line applies
# 2) An action.
#
# The two fields must be separated by one or more tabs or spaces.
#
# format:
#
# <msg_src_list>               <destination>
#
# where <msg_src_list> is a semicolon separated list of <facility>.<priority>
# where:
#
# <facility> is:
#      * - all (except mark)
#      kern,user,mail,daemon, auth, syslog, lpr, news, uucp, cron, authpriv, local0 - local7
#
# <priority or level> is one of (from high to low):
#      emerg,alert,crit,err(or),warn(ing),notice,info,debug
#      (meaning all messages of this priority or higher)
#
# <destination> is:
#      /filename - log to this file
#      username[,username2...] - write to user(s)
#      @hostname - send to syslogd on this machine
#      * - send to all logged in users
#
# example:
# "mail messages, at debug or higher, go to Log file. File must exist."
# "all facilities, at debug and higher, go to console"
# "all facilities, at crit or higher, go to all users"
#  mail.debug          /usr/spool/mqueue/syslog
#  *.debug             /dev/console
#  *.crit                      *

#   syslog messages with facilty / priority values of LOG_USER,   LOG_ALERT
user.alert            /tmp/syslog_user_alert

#   syslog messages with facilty / priority values of LOG_NEWS,  LOG_INFO
news.info           /tmp/syslog_news_info
```

*Figure 7. syslog.conf Configuration File*

## Configuring the 2216 for Remote Logging

To configure a 2216

1. In talk 6, configure the remote-logging facility as shown in Figure 8 on page 128. The IP address specified as the *source-ip-addr* should be an IP address that is configured in the 2216 for easier identification when the IP address or the hostname is shown in the remotely-logged ELS message. You should also verify that this IP address resolves quickly into a hostname by the name server or that

the name server at least responds quickly with “address not found.” To determine whether this happens, issue the **host** command on your workstation as follows:

```
workstation> host 5.1.1.1
host: address 5.1.1.1 NOT FOUND
workstation>
```

If the response takes more than 1 second, select an IP address which resolves more quickly.

2. In talk 6 configure events and subsystems for remote-logging, as shown in Figure 9 on page 129.
3. Write the configuration and reload the 2216.

```
ELS config>set remote source-ip-addr 5.1.1.1
Source IP Addr = 5.1.1.1

ELS config>set remote remote-ip-addr 192.9.200.1
Remote Log IP Addr = 192.9.200.1

ELS config>set remote local-id ** IBM/2216 **
Remote Log Local ID = ** IBM/2216 **

ELS config>set remote no-msgs-in-buffer 100
Number of messages in Remote Log Buffer must be 100-512
Number of Messages in Remote Buffer = 100

ELS config><B>set remote facility log_news
Default Syslog Facility = LOG_NEWS

ELS config>set remote level log_info
Default Syslog Level = LOG_INFO

ELS config>set remote on
Remote Logging is ON

ELS config>list remote

------------------   Remote Log Status  -----------------

Remote Logging is ON
Source IP Address = 5.1.1.1
Remote Log IP Address = 192.9.200.1
Default Syslog Facility = LOG_NEWS
Default Syslog Priority Level = LOG_INFO
Number of Messages in Remote Log =  100
Remote Logging Local ID = ** IBM / 2216 **
ELS config>
```

*Figure 8. Configuring the 2216 for Remote Logging*

```
ELS config>display sub snmp all
ELS config>remote sub snmp all log_news log_info

ELS config>display event srt.017
ELS config>remote event srt.017 log_news log_info

ELS config>display event stp.016
ELS config>remote event stp.016 log_user log_info

ELS config>display event stp.026
ELS config>remote event stp.026 log_news log_info

ELS config>display event stp.024
ELS config>remote event stp.024 log_news log_info

ELS config>display event ip.068
ELS config>remote event ip.068 log_news log_info

ELS config>display event ip.058
ELS config>remote event ip.058 log_news log_info

ELS config>display event ip.022
ELS config>remote event ip.022 log_news log_info

ELS config>display event gw.022
ELS config>remote event gw.22 log_news log_info

ELS config>display event arp.011
ELS config>remote event arp.011 log_user log_alert

ELS config>display event arp.002
ELS config>remote event arp.022 log_user log_alert

ELS config>list status
Subsystem:    SNMP
Disp levels:  ERROR INFO TRACE
Trap levels:  none
Trace levels: none
Remote levels:  ERROR INFO TRACE
        Syslog Facility/Level: LOG_NEWS LOG_INFO

Event     Display Trap    Trace     Remote
SRT.017   On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
STP.016   On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
STP.026   On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
STP.024   On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
IP.068    On      Unset   Unset
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
IP.058    On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
IP.022    On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
GW.022    On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_NEWS LOG_INFO
ARP.011   On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_USER LOG_ALERT
ARP.002   On      Unset   Unset     On
                                    Syslog Facility/Level: LOG_USER LOG_ALERT
```

*Figure 9. Configuring Subsystems and Events for Remote Logging*

## Remote Logging Output

Figure 10 on page 130 shows a sample from the /tmp/syslog_news_info file. Notice that the first message has a sequence number of 310. This means that the first 309 ELS messages were not sent from the source 2216. There are several reasons for this:

- The remote-logging facility had not completed initialization when the messages were first passed to ELS

- A route from the source 2216 to the remote workstation was not in the routing table
- The interface for the outbound UDP packet containing the ELS messages was not in the "Up" state

Notice in 1 that messages 311-313 did not get remote-logged. This is because an ARP request was outstanding and until the ARP response is received, all but the first packet is dropped in the source 2216. Additionally, the ARP cache is cleared at a user-configured refresh rate, and a new ARP request is issued. To determine when this is occurring, you can remote log events ARP.002 and ARP.011 in addition to the primary ELS events of interest. Figure 12 on page 132 shows ARP events logged to the *syslog_user_alert* file that account for events 445 and 446, which were indicated as missing in Figure 10.

```
Nov 20 12:03:16 workst01 root:  THIS IS A TEST MESSAGE  (news.info)
Nov 20 12:08:48 5.1.1.1 Msg [0310] from ** IBM / 2216 **: els:  IP.022: add nt 192.9.200.0 int 192.9.200.20
nt 0 int Eth/0

1 ( messages 311, 312, and 313 did not get remote-logged due to ARP request outstanding - see
  explanation in the text)

2 (messages 314 and 315 were logged to a separate
file - see explanation in the text)

Nov 20 12:08:48 5.1.1.1 Msg [0316] from ** IBM / 2216 **: els:  IP.068: routing cache cleared
Nov 20 12:08:48 5.1.1.1 Msg [0317] from ** IBM / 2216 **: els:  IP.022: add nt 5.0.0.0 int 5.1.1.1 nt 5 int Eth/4
Nov 20 12:08:48 5.1.1.1 Msg [0318] from ** IBM / 2216 **: els: SRT.017: Enabling SRT on port 5 nt 5 int Eth/4


(message 319 was logged to a separate file)

Nov 20 12:08:48 5.1.1.1 Msg [0320] from ** IBM / 2216 **: els:  IP.068: routing cache cleared

(120 messages not shown)

Nov 20 12:13:33 5.1.1.1 Msg [0441] from ** IBM / 2216 **: els:  GW.022: Nt fld slf tst nt 3 int Eth/3
Nov 20 12:13:33 5.1.1.1 Msg [0442] from ** IBM / 2216 **: els:  GW.022: Nt fld slf tst nt 6 int Eth/5
Nov 20 12:13:38 5.1.1.1 Msg [0443] from ** IBM / 2216 **: els:  GW.022: Nt fld slf tst nt 11 int ISDN/0


(messages 444 and 447 were logged to a separate file)


(messages 445 and 446 did not get remote-logged due to ARP request outstanding)

Nov 20 12:13:50 5.1.1.1 Msg [0448] from ** IBM / 2216 **: els:  GW.022: Nt fld slf tst nt 4 int ATM/0
Nov 20 12:13:50 5.1.1.1 Msg [0449] from ** IBM / 2216 **: els:  IP.068: routing cache cleared
Nov 20 12:13:50 5.1.1.1 Msg [0450] from ** IBM / 2216 **: els:  IP.058: del nt 4.0.0.0 rt via 0.0.0.4 nt 4 int ATM/0
```

*Figure 10. Sample Contents from Syslog News Info File*

If the initial ELS messages that are generated during and immediately after booting are of particular interest, then it is recommended that these messages also be displayed in the monitor queue, which is viewed with talk 2. Figure 11 on page 131 shows the talk 2 output including the initial messages that did not get remote-logged. Note that there is a message in the talk 2 output that indicates that the remote-logging facility is available. This does not indicate that a route exists to the remote workstation, nor that the associated interface is in the "Up" state. It simply provides a reference point before which no messages can be successfully remote-logged.

Also notice that you can account for the messages that were missing (indicated in Figure 10 with 2) in the talk 2 output.

```
12:08:17 SNMP.024: generic trc (P2) at snmp_mg.c(766): Now 0 trap destinations
12:08:17 SNMP.012: comm public added
12:08:17 SNMP.012: comm public added
12:08:17 SNMP.024: generic trc (P2) at lesConf.cpp(1491): Set  DEFAULT_ATMDEVNUM
= 4, DEFAULT_ATM_LINE_SPEED = 155
12:08:27 SNMP.022: ext err (Z1) at snmp_resconf.c(322): add_router_if_info(): sr
rdrec failed

12:08:27 SNMP.022: ext err (Z1) at snmp_resconf.c(322): add_router_if_info(): sr
rdrec failed

12:08:27 SNMP.028: err (E2) at snmp_moh.c(1583) : Duplicate
12:08:27 SNMP.028: err (E2) at snmp_moh.c(1583) : Duplicate
12:08:27 DOLOG: Found an ISDN interface record for ifn=0
12:08:27 DOLOG: ***************In config_mem_init
12:08:27  DOLOG: .....Remote Logging Facility is now available.....
12:08:28   GW.022: Nt fld slf tst nt 13 int PPP/3
12:08:28   IP.022: add nt 4.0.0.0 int 4.1.1.1 nt 4 int ATM/0

     ( 297 messages not shown )                                    Corresponding Sequence
                                                                   Numbers in
12:08:43   GW.022: Nt fld slf tst nt 12 int PPP/2                  Remote-Logging Files :
12:08:43   GW.022: Nt fld slf tst nt 13 int PPP/3
12:08:48   IP.022: add nt 192.9.200.0 int 192.9.200.20 nt 0 int Eth/0   [0310] first message logged
12:08:48  SRT.017: Enabling SRT on port 1 nt 0 int Eth/0          -- not logged (ARP request) --
12:08:48  STP.016: Select as root TB-1, det topol chg            -- not logged (ARP request)--
12:08:48  STP.026: Root TB-1, strt hello tmr                     -- not logged (ARP request)--
12:08:48  ARP.002: Pkt in 1 1 800 nt 0 int Eth/0                 [0314]
12:08:48  ARP.002: Pkt in 2 1 800 nt 0 int Eth/0                 [0315]
12:08:48   IP.068: routing cache cleared                         [0316]


     ( 126 messages not shown )

12:13:38   GW.022: Nt fld slf tst nt 11 int ISDN/0               [0443]
12:13:47  ARP.011: Del ent 1 3 nt 0 int Eth/0                    [0444]
12:13:47  ARP.011: Del ent 1 3 nt 0 int Eth/0                    -- not logged (ARP request) --
12:13:47  ARP.002: Pkt in 1 1 800 nt 5 int Eth/4                 -- not logged (ARP request)--
12:13:47  ARP.002: Pkt in 2 1 800 nt 0 int Eth/0                 [0447]
12:13:50   GW.022: Nt fld slf tst nt 4 int ATM/0                 [0448]
```

*Figure 11. Output from Talk 2*

You can use the timestamp, which appears in both the remote-logging output file and the talk 2 output, to determine when the first ELS message does get successfully remote-logged. To use the timestamp for this purpose, configure ELS such that the timestamp in the monitor queue displays the time-of-day.

Also notice in Figure 10 on page 130 that messages 311-313 did not get remote-logged. This is because an ARP request was outstanding and until the ARP response is received, all but the first packet is dropped in the source IBM 2216. The ARP cache is cleared at a user-configured refresh rate, and the device issues a new ARP request. To determine when ARP requests are occurring, events ARP.002 and ARP.011 can be remote-logged, in addition to the ELS events of interest. Figure 12 on page 132 shows ARP events logged to the *syslog_user_alert* file that account for events 445 and 446, which were indicated as missing in Figure 10 on page 130 .

```
Nov 20 12:02:53 worksta01 root: THIS IS A TEST MESSAGE (user.alert)
Nov 20 12:08:48 5.1.1.1 Msg [0314] from ** IBM / 2216 **: els:  ARP.002: Pkt in 1 1 800 nt 0 int Eth/0
Nov 20 12:08:48 5.1.1.1 Msg [0315] from ** IBM / 2216 **: els:  ARP.002: Pkt in 2 1 800 nt 0 int Eth/0
Nov 20 12:08:48 5.1.1.1 Msg [0319] from ** IBM / 2216 **: els:  ARP.002: Pkt in 2 1 800 nt 0 int Eth/0
Nov 20 12:13:47 5.1.1.1 Msg [0444] from ** IBM / 2216 **: els:  ARP.011: Del ent 1 3 nt 0 int Eth/0
Nov 20 12:13:47 5.1.1.1 Msg [0447] from ** IBM / 2216 **: els:  ARP.002: Pkt in 2 1 800 nt 0 int Eth/0
```

*Figure 12. Sample Contents from Syslog_user_alert File*

You can prevent the loss of ELS messages caused by this ARP sequence by establishing a static relationship between the IP address and the MAC address. The basic steps are outlined below and are illustrated in Figure 13 on page 133.

1. In talk 5, “ping” the remote workstation’s IP address
2. In talk 5, determine the interface (net) number used to send messages to the remote-workstation’s IP address
3. Use the net number from the previous step to determine the associated MAC address
4. In talk 6, add an ARP entry to establish a static IP address to MAC address relationship

```
*t 5
+p ip

IP>ping 192.9.200.1
PING 192.9.200.20 -> 192.9.200.1: 56 data bytes, ttl=64, every 1 sec.
56 data bytes from 192.9.200.1: icmp_seq=0. ttl=64. time=0. ms
----192.9.200.1 PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms

IP>dump

  Type   Dest net            Mask          Cost    Age       Next hop(s)
.
   Dir*  192.9.200.0         FFFFFF00      1       102305    Eth/0
.
IP>exit
+int
                                                    Self-Test  Self-Test   Maintenance
Net   Net'   Interface  Slot-Port                   Passed     Failed      Failed
0     0      Eth/0      Slot: 1   Port: 1               1          0           0
.
+p arp
ARP>dump
Network number to dump [0]? 0
Hardware Address        IP Address        Refresh
02-60-8C-2D-69-5D       192.9.200.1       2

Ctrl-P
*t 6
config>p arp
ARP config>add entry
Interface Number [0]? 0
Protocol [IP]? IP
IP Address [0.0.0.0]? 192.9.200.1
Mac Address []? 02608C2D695D
ARP config> list entry

Mac address translation configuration

IF #      Prot #  Protocol -> Mac address
   0           0  192.9.200.1 -> 02608C2D695D
ARP config>exit
Config>write

Ctrl-P

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

 (after reload, static ARP entry is active)
```

*Figure 13. Example of Setting Up a Static ARP Entry*

# Additional Considerations

## ELS Messages Containing IP Addresses

ELS messages containing an IP address which matches the IP address of the remote workstation will not be remote-logged, even if configured for remote-logging, and may appear under talk 2. These messages are discarded instead of being remote-logged in order to prevent excessive UDP packets from being sent on the network.

## Duplicate Logging

If a facility value is repeated in *syslog.conf*, for example:

```
user.debug        /tmp/syslog_user_debug
user.alert        /tmp/syslog_user_alert
```

The syslog daemon will log *user.debug* messages only to the */tmp/syslog_user_debug* file while user.alert messages will be logged to both the */tmp/syslog_user_debug* file and the */tmp/syslog_user_alert* file. This is consistent with the syslog design that logs the more severe conditions in multiple places.

To prevent this duplicate logging, it is recommended that different facility values be specified in the *syslog.conf* file. A total of 19 facility values are available.

## Recurring Sequence Numbers in Syslog Output Files

Depending upon the configuration of your network, it is possible for duplicate UDP packets containing ELS messages to arrive at the remote host. It is also possible for the packets to arrive in a different order than they were transmitted. An example of this phenomenon is shown in Figure 14. Notice that the messages with sequence numbers 628 through 633 are logged twice. Also notice that after the first occurrence of sequence number 0630, sequence number 0629 occurs again, followed by the second occurrence of 0630.

```
Apr 01 10:48:33 0.0.0.0 Msg [0628] from: RA22: : els: IPX.018: SAP gen rply sent nt 5 int TKR/1, 1 pkts
Apr 01 10:48:33 0.0.0.0 Msg [0628] from: RA22: : els: IPX.018: SAP gen rply sent nt 5 int TKR/1, 1 pkts
Apr 01 10:49:08 0.0.0.0 Msg [0629] from: RA22: : els: IPX.037: RIP resp sent nt 0 int TKR/0, 1 pkts
Apr 01 10:49:08 0.0.0.0 Msg [0630] from: RA22: : els: IPX.018: SAP gen rply sent nt 0 int TKR/0, 1 pkts
Apr 01 10:49:08 0.0.0.0 Msg [0629] from: RA22: : els: IPX.037: RIP resp sent nt 0 int TKR/0, 1 pkts
Apr 01 10:49:08 0.0.0.0 Msg [0630] from: RA22: : els: IPX.018: SAP gen rply sent nt 0 int TKR/0, 1 pkts
Apr 01 10:49:33 0.0.0.0 Msg [0631] from: RA22: : els: IPX.037: RIP resp sent nt 5 int TKR/1, 1 pkts
Apr 01 10:49:33 0.0.0.0 Msg [0631] from: RA22: : els: IPX.037: RIP resp sent nt 5 int TKR/1, 1 pkts
Apr 01 10:49:33 0.0.0.0 Msg [0632] from: RA22: : els: IPX.018: SAP gen rply sent nt 5 int TKR/1, 1 pkts
Apr 01 10:49:33 0.0.0.0 Msg [0632] from: RA22: : els: IPX.018: SAP gen rply sent nt 5 int TKR/1, 1 pkts
Apr 01 10:50:08 0.0.0.0 Msg [0633] from: RA22: : els: IPX.037: RIP resp sent nt 0 int TKR/0, 1 pkts
Apr 01 10:50:08 0.0.0.0 Msg [0633] from: RA22: : els: IPX.037: RIP resp sent nt 0 int TKR/0, 1 pkts
```

*Figure 14. Example of Recurring Sequence Numbers in Syslog Output*

Because neither Syslog nor UDP has the ability to handle duplicate or out of sequence packets, it is important to recognize the possibility of duplicate sequence numbers occurring.

# Chapter 13. Configuring and Monitoring the Event Logging System (ELS)

This chapter describes how to configure events logged by ELS and how to use the ELS commands. The information includes the following sections:

- "Accessing the ELS Configuration Environment"
- "ELS Configuration Commands"
- "Entering and Exiting the ELS Operating Environment" on page 155
- "ELS Monitoring Commands" on page 156

For more information on the Event Logging System and how to interpret ELS event messages, refer to "Chapter 12. Using the Event Logging System (ELS)" on page 117.

## Accessing the ELS Configuration Environment

The ELS configuration environment is characterized by the `ELS config>` prompt. Commands entered at this prompt are described "Chapter 13. Configuring and Monitoring the Event Logging System (ELS)".

To enter the ELS configuration environment:

1. Enter **talk 6**.

   The monitoring displays the `Config>` prompt. If the prompt does not appear, press **Return**.
2. At the `Config>` prompt, enter the following command to access ELS:

   ```
   event
   ```

   The monitoring displays the ELS configuration prompt (`ELS config>`). Now, you can enter ELS configuration commands.

To leave the ELS configuration environment, enter the **exit** command.

## ELS Configuration Commands

Table 14 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 14. ELS Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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. |

*Table 14. ELS Configuration Command Summary (continued)*

| Command | Function |
|---|---|
| Delete | Deletes an event number from an existing group or deletes an entire group. |
| Display | Enables message display on the console monitor. |
| Filter | Filter ELS messages based upon the net number. |
| List | Lists information on ELS settings and messages. |
| Nodisplay | Disables message display on the console. |
| Noremote | Disables remote logging to a remote workstation. |
| Notrace | Controls disablement of packet trace events. |
| Notrap | Keeps messages from being sent out in SNMP traps. |
| Remote | Allows messages to be logged to a remote workstation. |
| 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. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**add** *group_name subsystem.event_number*

**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:**

**clear**

**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:**

| | |
|---|---|
| **default** | display |
| | trap |
| | remote |

**display** *event* **OR** *group* **OR** *subsystem*
Controls the output of the display of messages to the monitoring.

**trap** *event* **OR** *group* **OR** *subsystem*
Controls the generation of traps to the network management station.

**remote** *event* **OR** *group* **OR** *subsystem*
Controls the generation of traps to the remote station.

## 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:**

| | |
|---|---|
| **delete** | *group_name subsystem.event_number* |

## Display

Use the **display** command to enable message displaying on the monitoring monitor for specific events, a range of events for a subsystem, groups, or subsystems.

**Syntax:**

| | |
|---|---|
| **display** | event . . . |
| | group . . . |
| | range . . . |
| | subsystem . . . |

**event** *subsystem.event#*
Displays messages of the specified event (*subsystem.event#*).

**group** *groupname*
Displays messages of a specified group (*groupname*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event in the specified event range.

Displays a range of messages for the specified subsystem.

**Example:**

```
display range gw 19 22
```

Displays events gw.19, gw.20, gw.21, and gw.22.

**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<br><br>**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 monitoring. Never display all subsystems when you are communicating with the router through a remote monitoring. This causes the router to spend most of its time communicating with the remote monitoring. |
| **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 |
| **DIAL** | Dial circuits |
| **DLS** | Data Link Switching |
| **DN** | DECnet |
| **DNAV** | DNA Phase V |
| **DVM** | DVMRP Multicast Routing Protocol |
| **ENCR** | Data Encryption |
| **ESC** | ESCON |
| **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 |
| **ILMI** | ATM Interim Local Management Interface |
| **LCS** | Logical Channel Station |
| **LEC** | ATM LAN Emulation Client |
| **LECS** | LAN Emulation Configuration Server |
| **LES** | LAN Emulation Server |
| **LLC** | Logical Link Control |
| **LSA** | Link Services Architecture |
| **LSI** | LAN Switch Integration |
| **LNM** | LAN Network Manager |
| **MCF** | MAC Filtering |
| **MPC** | Multi-Path Channel |
| **MSPF** | OSPF Multicast extensions |
| **NBS** | NetBIOS Support Subsystem |
| **NOT** | Non-supported Protocol Forwarder |
| **OSPF** | Open SPF-based Routing Protocol |
| **PCA** | Parallel Channel Adapter |
| **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.25bis |
| **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 |

## Filter

Use the **filter** command to access the filter configuration command environment. See “ELS Net Filter Configuration Commands” on page 153 for complete command details.

**Syntax:**

**filter** net

## List

Use the **list** command to get updated information regarding ELS settings and listings of selected messages.

**Syntax:**

**list** all
filter-status
groups
pin
remote-log status
status
subsystem . . .
subsystems all
trace-status

**all** Lists information from all the **list** categories.

**filter-status**
Lists ELS net number filters.

**groups**
Lists the user-defined group names and contents.

**pin** Lists the current number of ELS event messages sent in SNMP traps (per second).

**remote-log status**
Lists the current values of remote logging options.

**Example:**

```
list r

Remote Logging is ON
Source IP Address = 192.67.38.2
Remote Log IP Address = 192.9.200.1
Default Syslog Facility = LOG_DAEMON
Default Syslog Priority Level = LOG_CRIT
Number of Messages in Remote Log = 256
Remote Logging Local ID = MYHOSTNAME
```

**status** Lists the subsystems, groups, and events that have been modified by the **display**, **nodisplay**, **trap**, and **notrap**, **trace**, **notrace**, **remote**, and **noremote** commands.

**Example:**

```
list status

Subsystem:             TKR
Disp Levels:           STANDARD
Trap levels:           none
Trace levels:          none
Remote levels:         ERROR INFO TRACE
Syslog Facility/Level: LOG_USER LOG_INFO

Group        Disp     Trap    Trace  Remote
Mygroup     Unset    Unset    Unset    On
                                      Syslog Facilty/Level: LOG_DAEMON LOG_CRIT

Event        Disp     Trap    Trace  Remote
IP.007      Unset    Unset    Unset    On
                                      Syslog Facility/Level: LOG_CRON LOG_NOTICE
```

**Note:** Not only is remote logging enabled, but the display includes the Syslog Facility/Level values for each subsystem, group, and event. Ranges of events are listed as individual events.

**subsystem**
Lists names, events, and descriptions of all subsystems.

(Example output from a **list subsystem** command can be found beginning on page 159.)

**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 all**
Lists all events in all subsystems.

**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.

**Syntax:**

**nodisplay** event. . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Suppresses the displaying of a specified event (*subsystem.event#*).

**group** *groupname*
Suppresses the displaying of messages that were previously added to the specified group (*groupname*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Suppresses the displaying of a range of messages for the specified subsystem.

**Example:**

```
nodisplay range gw 19 22
```

Suppresses the display of events gw.19, gw.20, gw.21, and gw.22.

**subsystem** *subsystemname*
Suppresses the displaying of messages associated with the specified subsystem.

# Noremote

Use the **noremote** command to suppress the logging of events to a remote workstation based on event number, group, range of events, or subsystem.

**Note:** With the **noremote** command, there is usually no need to specify a *syslog_facility* and *syslog_level*, such as there is with the **remote** command. However, for **noremote subsystem** command, there exists the option of selectively suppressing specific message levels (for example, “error” only or “trace” only) rather than turning them all off. (If you do not specify any particular message level, “all” is assumed). Additionally, with the **noremote subsystem** command, you can set a *syslog_facility* and *syslog_level* for any remaining message levels that have not been turned off.

**Syntax:**

**noremote** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Suppresses the remote logging of messages for the specified event.

**group** *group.name*
Suppresses the remote logging of messages that were previously added to the specified group (*group.name*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Suppresses the remote logging of a range of messages for the specified subsystem.

**Example:**

```
noremote range gw 19 22
```

Suppresses the remote logging of events gw.019, gw.020, gw.021, and gw.022

**subsystem** *subsystem.name [syslog_facility syslog_level]*
Suppresses the remote logging of messages associated with the specified subsystem (*subsystem.name*).

**Example 1:**

```
noremote subsystem tkr
```

Suppresses the remote logging of all “tkr” messages.

**Example 2:**

```
ELS config> noremote subsystem tkr info
ELS config> SYSLOG FACILITY[LOG_USER]?
ELS config> SYSLOG LEVEL[LOG_INFO]?
```

In this example, “LOG_USER” and “LOG_INFO” were the values last picked for subsystem TKR. The command specified turns off the remote logging for subsystem TKR only for messages coded for “info”. Because *syslog_facility* and *syslog_level* was not specified, the software prompts for *syslog_facility* and *syslog_level*. If you enter another value at the prompts, that value will replace *syslog_facility* and *syslog_level* for the remaining remote-logged messages for the TKR subsystem.

Use the **list all** or **list status** commands to display what you have set with the **noremote** and **remote** commands.

For more information about *syslog_facility* and *syslog_level* see “Remote” on page 145 .

## Notrace

Disables packet trace for the specified event/range/subsystem/group.

**Syntax:**

**notrace** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Suppresses the sending of packet trace data for the specified event#

**group** *groupname*
Suppresses the sending of packet trace data that was previously added to the specified group (groupname).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Disables the sending of packet trace data for a range of messages for the specified subsystem.

**Example:**

```
trace range gw 19 22
```

Suppresses the sending of packet trace data for events gw.19, gw.20, gw.21, and gw.22.

**subsystem** *subsystemname*
Suppresses the sending of packet trace data for the specified subsystem (subsystemname).

## 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:**

**notrap** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Suppresses the sending of the specified message in an SNMP trap (*subsystem.event#*).

**group** *groupname*
Suppresses the sending of messages in SNMP traps that were previously added to the specified group (*groupname*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Suppresses the sending of messages for the events in the specified range for the specified subsystem in SNMP traps.

**Example:**

```
notrap range gw 19 22
```

Suppresses the sending of messages for events gw.19, gw.20, gw.21, and gw.22 in SNMP traps.

**subsystem** *subsystemname*
Suppresses the sending of messages in SNMP traps that are associated with the specified subsystem.

## Remote

Use the **remote** command to select the events to be logged to a remote workstation by event number, range of events, group, or subsystem.

**Syntax:**

**remote** event . . .
range . . .
group . . .
subsystem . . .

**event** *subsystem.event# syslog_facility syslog_level*
Causes the specified event to be logged remotely.

Syslog facility and level values are used by the syslog daemon in the remote workstation to determine where to log the messages. This value overrides the default values that are set with the **set facility** and **set level** commands.

*syslog_facility*
log_auth
log_authpriv
log_cron
log_daemon
log_kern
log_lpr
log_mail
log_news
log_syslog

log_user
log_uucp
log_local0-7

*syslog_level*

log_emerg
log_alert
log_crit
log_err
log_warning
log_notice
log_info
log_debug

These values do NOT have any particular association with any daemons on the IBM 2216. They are merely identifiers which are used by the syslog daemon on the remote workstation.

**range** *subsystemname first_event_number last_event_number syslog_facility syslog_level*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_numbe*r is the number of the last event of the specified event range.

Causes the events in the specified range for the specified subsystem to be remotely logged based on the *syslog_facility* and *syslog_level* values. See “the remote event command” on page 145.

**Example:**

```
remote range gw 19 22 log_user log_info
```

Causes the event gw.19, gw.20, gw.21, and gw.22 to be logged remotely on the *syslog_facility* value of log_user and the *syslog_level* value of log_info.

**group** *group.name syslog_facility syslog_level*

Allows events belonging to the specified group to be logged remotely based on the *syslog_facility* and *syslog_level* values. See “the remote event command” on page 145.

**subsystem** *subsystem.name message_level syslog_facility syslog_level*

Where *subsystem.name* is the name of the subsystem and *message_level* is the level of messages selected in the subsystem.

Causes the events within the specified *subsystem.name* whose *message_level* agrees with the specified *message_level* to be logged remotely at the files based on the *syslog_facility* and *syslog_level* values. See “the remote event command” on page 145.

*Message_level* is a value such as “ALL,” “ERROR,” “INFO,” or “TRACE” . See “Logging Level” on page 119. The value specified in the **remote** command must agree with the value as coded on the particular event within the subsystem, or that event within the subsystem will not be remotely logged.

**Example:**

```
remote subsystem TKR all log_user log_info
```

In the above example, all messages in subsystem TKR ("all" includes any messages coded for "error," "info," or "trace") will be logged remotely based on log_user and log_info values at the remote host.

Use the **list all** or **list status** commands to display what you have set with the **noremote** and **remote** commands.

## 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:**

| | |
|---|---|
| **set** | pin . . . |
| | remote-logging . . . |
| | 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.)

**remote-logging**

Use the **set remote-logging** command to configure remote logging options. When these options are configured from the monitoring environment, the changes take effect immediately, and return to their previously configured settings when the device is rebooted.

**Syntax:**

| | |
|---|---|
| **set remote-logging** | on |
| | off |
| | facility . . . |
| | level . . . |
| | no-msgs |
| | remote_ip_addr . . . |
| | source_ip_addr ... |
| | local_id |

**on** Turns remote logging on. Remote logging is now enabled to allow any messages selected by the **remote** command to be actively logged.

**off** Turns remote logging off. All messages selected by the 'remote' command will be prevented from being logged.

**facility**

Specifies a value that, in combination with the *level* value, is used by the syslog daemon in the remote workstation to determine where to log messages. This value is used for all remotely-logged ELS messages unless you specify a different value for a particular ELS event, range, group, or subsystem with the **remote** command.

These are all possible syslog facility values:

log_auth
log_authpriv
log_cron
log_daemon
log_kern
log_lpr
log_mail
log_news
log_syslog
log_user
log_uucp
log_local0-7

**level** Specifies a value that, in conjunction with the *facility* value, is used by the syslog daemon in the remote workstation to determine where to log messages. This value is used for all remotely-logged ELS messages unless you specify a different value for a particular ELS event, range, group, or subsystem with the **remote** command.

These are all possible syslog level values:

log_emerg
log_alert
log_crit
log_err
log_warning
log_notice
log_info
log_debug

**no-msgs**
Specifies the number of messages in the buffer for the remote log before log wraps.

**remote_ip_addr**
This is an ip address of the form xxx.xxx.xxx.xxx where xxx can be any integer 0 to 255. It represents the ip address of the remote host where the log files reside.

**source_ip_addr**
This is an ip address of the form xxx.xxx.xxx.xxx where xxx can be any integer 0 to 255.

You should use an IP address that is configured in the 2216 for easier identification when the IP address or the hostname is shown in the remotely-logged ELS message. You should also verify that this IP address is quickly resolved to a hostname by the name server, or at least that the name server responds quickly with "address not found."

To determine that the IP address resolves properly enter the **host** command on your workstation as shown:

```
workstation>host 5.1.1.1
host: address 5.1.1.1 NOT FOUND
workstation>
```

If the response takes more than 1 second, select an IP address that resolves more quickly.

**local_id**
This is any character string of up to 32 characters which is included in the logged message at the remote file and can help identify which machine logged the message.

**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.

**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 environment, the changes take effect immediately, and return to their previously configured settings when the device is rebooted.

**Note:** Tracing should be used only under the direction of trained support personnel. Tracing, especially when used with disk-shadowing enabled, uses device resources and can impact overall performance and throughput.

**Syntax:**

**set trace**
- decode
- default-bytes-per-pkt
- disk-shadowing
- max-bytes-per-pkt
- 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 or on] or record-size or time-limit or delete-file or max-file-size]**
Turns disk shadowing on or off, sets the maximum trace file size, or sets the maximum time for disk-shadowing traces.

**[off or 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 monitoring.

**Note:** Disk shadowing should be set to OFF whenever the WRITE, TFTP software, RETRIEVE system dump, or COPY software commands are issued.

**disk-shadowing delete-file**
Deletes the trace file.

**disk-shadowing max-file-size** *Mbytes*
Sets the maximum file size for the trace file.

**Valid Values**: 1 Mbyte to 16 Mbytes

**Default Value**: 10 Mbytes

**disk-shadowing record-size** *bytes*
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.

**disk-shadowing time-limit** *hours*
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.

**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.

**Valid Values:** 0, ≥10,000

**Default Value:** 0

**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.)

**wrap-mode [off or 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.

## Trace

Enables packet trace for the specified event/range/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.

**Syntax:**

**trace** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Causes the specified trace event (*subsystem.event#*) to be displayed on the system monitoring.

**group** *groupname*
Allows trace events that were previously added to the specified group to be displayed on the router monitoring.

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Causes the trace events in the specified range for the specified subsystem to be displayed on the system monitoring.

**Example:**

```
trace range gw 19 22
```

Causes the trace events gw.19, gw.20, gw.21, and gw.22 to be displayed on the system monitoring.

**subsystem** *subsystemname*
Allows trace events associated with the specified subsystem to be displayed on the router monitoring.

## 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** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Causes the specified message (*subsystem.event#*) to be sent to a network management workstation in an SNMP trap.

**group** *groupname*
Allows messages that were previously added to the specified group to be sent to a network management workstation in an SNMP trap.

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Causes the messages that are in the specified range for the specified subsystem to be sent to a network management workstation in an SNMP trap.

**Example:**

```
trap range gw 19 22
```

Causes the messages in events gw.19, gw.20, gw.21, and gw.22 to be sent to a network management workstation in an SNMP trap.

**subsystem** *subsystemname*
Allows messages associated with the specified subsystem to be sent to a management station in an SNMP trap.

**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.

# ELS Net Filter Configuration Commands

ELS net filters give you the capability of looking only at ELS messages with certain net numbers and discarding other ELS messages.

When you create a filter, you specify the subsystem, event, or range of events to which the filter applies. You also specify the queue (for example, “DISPLAY”, “TRAP”, “TRACE”, or “REMOTE-LOGGING”). Finally, you specify the net number (or range of net numbers) that you want to filter.

When you enable the filter, messages that have been turned on by the ELS commands are subject to filtering. The filter allows only messages with the specified net numbers. The filter causes the device to discard messages that do not contain the specified net numbers.

By reducing the number of ELS messages sent, you can more easily locate messages for the interfaces in which you are interested.

This section describes the commands to configure the ELS net filters. To configure these filters, enter the **filter net** command at the ELS> prompt. Then, enter the configuration commands at the `ELS Filter net>` prompt.

*Table 15. ELS Net Filter Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Create | Creates a filter and assigns it a number. A maximum of 64 filters is allowed. |
| Delete | Deletes a specified filter number or all filters. |
| Disable | Disables a specified filter number or all filters. |
| Enable | Enables a specified filter number or all filters. |
| List | Lists a specified filter number or all filters. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Create

Use the **create** command to create an ELS net filter.

**Syntax:**

**create** ***queue*** event *event_name net#_start net#_end*
range *event_range net#_start net#_end*
subsystem *subsystem_name net#_start net#_end*

**queue** The queue for which you are setting the filter. The valid queues are:

Display
Trace
Trap
Remote

**event** *event_name net#_start net#_end*
Specifies the event and net numbers that you are filtering.

If you specify *net#_start* and *net#_end* as the same number, you are filtering on a single net number.

The command **create trap event GW.009 2 10** filters traps for message GW.009 for net numbers 2 through 10.

**range** *event_range net#_start net#_end*
Specifies the range of ELS messages and net numbers that you are filtering.

If you specify *net#_start* and *net#_end* as the same number, you are filtering on a single net number.

The command **create remote range ipx 19 22 3 6** filters all ipx messages beginning with IPX.019 and ending with IPX.022 for net numbers 3 through 6 for remote logging.

**subsystem** *subsystem_name net#_start net#_end*
Specifies the subsystem and net numbers that you are filtering.

If you specify *net#_start* and *net#_end* as the same number, you are filtering on a single net number.

The command **create display subsys ip 1 1**, filters all ELS messages for the ip subsystem that contain net number 1 to the display. All other ip subsystem messages are discarded.

## Delete

Use the **delete** command to delete a specific ELS filter or all ELS filters.

**Syntax:**

**delete** all
filter *filter#*

**all** Deletes all currently configured filters.

**filter** *filter#*
Deletes the filter specified by *filter#*. Use the **list** command to obtain the number for the filter you want to delete.

## Disable

Use the **disable** command to disable a specific ELS filter or all ELS filters.

**Syntax:**

**disable** all
filter *filter#*

**all** Disables all currently configured filters.

**filter** *filter#*
Disables the filter specified by *filter#*. Use the **list** command to obtain the number for the filter you want to disable.

## Enable

Use the **enable** command to enable a specific ELS filter or all ELS filters.

**Syntax:**

**enable** all
filter *filter#*

**all** Enables all currently configured filters.

**filter** *filter#*
Enables the filter specified by *filter#*. Use the **list** command to obtain the number for the filter you want to enable.

### List

Use the **list** command to list a specific ELS filter or all ELS filters.

**Syntax:**

**list** all
filter *filter#*

**all** Lists all currently configured filters.

**filter** Lists the filter specified by *filter#.*

## Entering and Exiting the ELS Operating Environment

The ELS monitoring 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 13. Configuring and Monitoring the Event Logging System (ELS)” on page 135.

To enter the ELS monitoring environment from OPCON:

1. Enter the **talk 5** command.

   ```
   * talk 5
   ```

   The monitoring displays the GWCON prompt (+). If the prompt does not appear when you first enter GWCON, press **Return**.
2. At the GWCON prompt, enter the following command to access ELS:

   ```
   + event
   ```

   The monitoring displays the ELS monitoring prompt (ELS>). Now, you can enter ELS monitoring commands.

To leave the ELS monitoring environment, enter the **exit** command.

# ELS Monitoring Commands

This section summarizes and then explains all the ELS monitoring commands. After accessing the ELS Monitoring environment, you can enter ELS monitoring commands at the ELS> prompt.

*Table 16. ELS Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Clear | Resets to zero the counts of messages associated with specified events, groups, or subsystems. |
| Display | Enables message display on the console. |
| Exit | Exits the ELS console process and returns the user to GWCON. |
| Filter | Filter ELS messages based upon the net number. |
| List | Lists information on ELS settings and messages. |
| Nodisplay | Disables message display on the console. |
| Noremote | Disables remote logging to file at remote workstation. |
| Notrace | Disables trace event display on the console. |
| Notrap | Keeps messages from being sent out in SNMP traps to the network management workstation. |
| Packet-trace | Provides an enhanced central environment for setting and listing active packet tracing parameters. |
| Remote | Allows messages to be logged at a file on a remote 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. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Clear

Use the **clear** command to reset to zero the counts of the display, trace, trap, or remote commands as they relate to specific events, groups or subsystems.

**Syntax:**

**clear** event . . .
group . . .
subsystem . . .

**event** *subsystem. event#*
Resets the count of events to zero for displaying, trapping, tracing or remote logging of the specified event (*subsystem.event#*).

**group** *group.name*
Resets the count of events to zero for displaying, trapping, tracing or remote logging of the specified group (*group.name*).

**subsystem** *subsystem.name*
Resets the count of events to zero for displaying, trapping, tracing or remote logging of the specified subsystem (*subsystem.name*).

## Display

Use the display command to enable the message display on the monitoring monitor for specific events.

**Syntax:**

**display** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Displays messages for the specified event (subsystem.event#).

**group** *groupname*
Displays messages of a specified group (*groupname*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event in the specified event range.

Displays a range of messages for the specified subsystem.

**Example:**

```
display range gw 19 22
```

Displays events gw.19, gw.20, gw.21, and gw.22.

**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.

## 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:**

**files trace tftp** active-bank ...
bank-a ...

bank-b ...

net-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).

## Filter

Use the **filter** command to access the filter configuration command environment. See “ELS Net Filter Monitoring Commands” on page 178 for complete command details.

**Syntax:**

**filter** net

## List

Use the list command to get updated information regarding ELS settings and to get listings of selected messages.

**Syntax:**

**list** all
active . . .
event . . .
filter-status
groups . . .
pin
remote-log status
subsystems . . .
trace-status

**all** Lists all subsystems, defined groups, enabled subsystems, enabled events, and pins.

**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
```

If Remote logging is turned on, those events displayed as active for a subsystem will have an “R” next to their name.

**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: source_ip_address -> destination_ip_address
Active:  Count: 84182
```

If Remote-logging had been activated for this event, and the *syslog_facility* and *syslog_level* values were log_daemon and log_crit, the last lines would look like:

```
Active:  R count:84182
Syslog Facility: log_daemon   Syslog Level: log_crit
```

**filter-status**
Lists ELS net number filters.

**groups** *group.name*
Displays the user-defined group names.

**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
```

**remote-log status**
Lists the current values of the remote logging options set in the **set remote-logging** command.

**Example:**

```
list r

Remote Logging is On
Source Ip Address = 192.9.200.8
Remote  Log IP Address = 192.9.200.1
Default Syslog Facility = LOG_USER
Default Syslog Priority Level = LOG_INFO
Number of Messages in Remote Log = 256
Remote Logging Local ID =  SPHINX
```

**subsystem** *subsystem.name*
Lists event names, the total number of events that have occurred, and their descriptions.

**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.

**Example:**

## ELS Monitoring Commands (Talk 5)

**list subsystem**

```
Name      Events  Description

ALL               All subsystems
GW           101  Router base and network library
FLT            7  Filter Library
BRS            5  Bandwidth  Reservation
ARP          142  Address Resolution Protocol
IP           100  Internet Protocol
ICMP          21  Internet Control Message Protocol
TCP           57  TCP
UDP            6  User Datagram Protocol
BTP           13  BOOTP relay agent
RIP           22  IP Routing Information Protocol
OSPF          73  Open SPF-Based Routing Protocol
MSPF          17  OSPF Multicast extensions
TFTP          29  TFTP Protocol
SNMP          28  Simple Network Management Protocol
DVM           21  DVMRP Multicast Routing Protocol
DN           115  DECnet
XN            21  XNS/IPX/DDS common processing
IPX          110  Internetwork Packet Exchange Protocol
CLNP          58  ISO 8473 - CLNP
ESIS          24  ISO 9542 - ESIS Protocol
ISIS          58  ISO 10589 - ISIS Protocol
DNAV          26  DNA Phase V
AP2           70  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           94  Source Routing Transparent Bridge
STP           32  Spanning Tree Protocol
BR            30  Bridge/Routing
SRLY          28  SDLC Relay
ETH           47  Ethernet Handler
SL            35  Serial Line Handler
TKR           45  Token Ring Handler
X25           53  X.25 Protocols
FDDI          27  FDDI Handler
SDLC          95  IBM SDLC
FRL           97  Frame Relay
PPP          186  Point-to-Point
X251          16  X.25-Physical-Layer
X252          34  X.25-Frame-Layer
X253          42  X.25-Packet-Layer
ISDN          43  Integrated Services Digital Network
IPPN           4  IP Protocol Net
WRS           33  WAN Restoral
LNM           60  LNM
LLC          168  Logical Link Control
BGP           74  Border Gateway Protocol
MCF            9  MAC Filtering
DLS          497  Data Link Switching
V25B          28  CCITT/ITU V.25bis
BAN           29  Boundary Access Node
COMP          26  Data Compression Engines
NBS           50  NetBIOS Support Subsystem
ATM          216  Asynchronous Transfer Mode
LEC          174  ATM LAN Emulation Client
APPN          28  Advanced Peer-to-Peer Networking
ILMI          23  ATM Interim Local Management Interface
SAAL          26  ATM Signalling ATM Adaptation Layer
SVC           26  ATM Signalling
LES          361  LAN Emulation Services
LECS         145  LAN Emulation Configuration Server
EVLOG          1  EventLog() error logging system
NOT           15  Forwarder messages not loaded
NHRP         211  Next Hop Resolution Protocol
XTP           58  X.25 Transport

ESC           67  ESCON Handler
PCA           67  PCA Handler
LCS           22  LCS Handler
LSA           61  LSA Handler
MPC           30  MPC Handler
SCSP          34  Server Cache Synchronization Protocol
ALLC          36  ATM LLC (RFC1483)
NDR           38  Network Dispatcher Router Feature
MLP           93  Multilink-PPP
SEC           30  Security Protocols
ENCR           4  Data Encryption Engines
PM             6  Presence Manager
DGW            9  Default Gateway
QLLC          54  QLLC-Packet-LayerName  Events  Description
VLAN          20  Virtual LAN
```

**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 packet_type source_Ethernet_address ->
                      destination_Ethernet_address nt network
ETH.002    UE-ERROR   rcv unkwn typ packet_type source_Ethernet_address ->
                      destination_Ethernet_address nt network
ETH.010    C-INFO     LLC unk SAP DSAP source_Ethernet_address ->
                      destination_Ethernet_address nt network
```

**subsystems all**

Lists all events, logging levels, and messages for every event that has occurred on the router.

**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.

**Syntax:**

**nodisplay** event . . .

group . . .

range . . .

subsystem . . .

**event** *subsystem.event#*
Suppresses the displaying of messages for the specified event.

**group** *group.name*
Suppresses the displaying of messages that were previously added to the specified group (*group.name*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Suppresses the displaying of a range of messages for the specified subsystem.

**Example:**

```
nodisplay range gw 19 22
```

Suppresses the display of events gw.19, gw.20, gw.21, and gw.22.

**subsystem** *subsystem.name*
Suppresses the displaying of messages associated with the specified subsystem (*logging level*).

## Noremote

Use the **noremote** command to select and turn off messages logging to a remote workstation.

**Syntax:**

**noremote** event . . .

group . . .

range . . .

subsystem . . .

**event** *subsystem.event#*
Suppresses the remote logging of messages for the specified event.

**group** *group.name*
Suppresses the remote logging of messages that were previously added to the specified group (*group.name*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Suppresses the remote logging of a range of messages for the specified subsystem.

**Example:**

```
noremote range gw 19 22
```

Suppresses the remote logging of events gw.19, gw.20, gw.21, and g.22

**subsystem** *subsystem.name*
Suppresses the remote logging of messages associated with the specified subsystem (*logging level*).

**Example:**

```
noremote subsystem tkr
```

**Note:** With Noremote, there is no need to specify a Syslog Facility and Level, such as there is with Remote.

Use the **list event** and **list active** commands to verify what you set with the **remote** and **noremote** commands.

## Notrace

Use the **notrace** command to stop display of selected trace events at the monitoring.

**Syntax:**

**notrace** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Suppresses the display of the specified tracing event.

**group** *groupname*
Suppresses the display of tracing events related to the specified group (*groupname*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Disables the sending of packet trace data for a range of messages for the specified subsystem.

**Example:**

```
notrace range gw 19 22
```

Suppresses the sending of packet trace data for events gw.19, gw.20, gw.21, and gw.22.

**subsystem** *subsystemname [logging-level]*
Suppresses the display of tracing events that are associated with the specified subsystem and logging level. If you do not specify a *logging-level* you suppress tracing for all logging levels for the subsystem.

**Example:**

```
notrace subsystem atm error
```

```
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:**

**notrap** event. . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Suppresses the sending of the specified message in an SNMP trap (*subsystem.event#*).

**group** *groupname*
Suppresses the sending of messages in SNMP traps that were previously added to the specified group (*groupname*).

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Suppresses the sending of messages for the events in the specified range for the specified subsystem in SNMP traps.

**Example:**

```
notrap range gw 19 22
```

Suppresses the sending of messages for events gw.19, gw.20, gw.21, and gw.22 in SNMP traps.

**subsystem** *subsystemname [logging-level]*
Suppresses the sending of messages in SNMP traps that are associated with the specified subsystem and logging level. If you do not specify a *logging-level* you suppress trapping for all logging levels for the subsystem.

**Example:**

```
notrap subsystem tkr error
```

## Packet Trace

Use the **packet-trace** command to display/enable/disable packet tracing information for various subsystems. This command provides function similar to the **Trace** command.

**Syntax:**

**packet-trace**

Use the **Exit** command when you are finished using Packet Trace.

For complete command descriptions, see “Packet-trace Monitoring Commands” on page 175 .

## Remote

Use the **remote** command to select the events to be logged to a remote file by event number, range of events, group, or subsystem.

**Syntax:**

**remote** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event# syslog_facility syslog_level*
Causes the specified event to be logged remotely.

Syslog facility and level values are used by the syslog daemon in the remote workstation to determine where to log the messages. This value overrides the default values that are set with the **set facility** and **set level** commands.

*syslog_facility*
log_auth
log_authpriv
log_cron
log_daemon
log_kern
log_lpr
log_mail
log_news
log_syslog
log_user
log_uucp
log_local0-7

*syslog_level*
log_emerg
log_alert
log_crit
log_err
log_warning
log_notice
log_info
log_debug

These values do NOT have any particular association with any daemons on the IBM 2216. They are merely identifiers which are used by the syslog daemon on the remote workstation.

**Example:**

```
remote event gw.019 log_user log_info
```

**group** *group.name syslog_facility syslog_level*

Allows events belonging to the specified group to be logged remotely based on the *syslog_facility* and *syslog_level* values. See “the remote event command” on page 165.

**range** *subsystemname first_event_number last_event_number syslog_facility syslog_level*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Causes the events in the specified range for the specified subsystem to be remotely logged based on the *syslog_facility* and *syslog_level*. See “the remote event command” on page 165.

**Example:**

```
remote range gw 19 22 log_user log_info
```

Causes the event gw.19, gw.20, gw.21, and gw.22 to be logged remotely to the files specified by the *syslog_facility* value of log_user and the *syslog_level* value of log_info.

**subsystem** *subsystem.name message_level syslog_facility syslog_level*

Where *subsystem.name* is the name of the subsystem and *message_level* is the level of messages selected in the subsystem.

Causes the events within the specified *subsystem.name* whose *message_level* agrees with the specified *message_level* to be logged remotely based on the *syslog_facility* and *syslog_level*. See “the remote event command” on page 165.

*Message_level* is a value such as “ALL,” “ERROR,” “INFO,” or “TRACE” . See “Logging Level” on page 119. The value specified in the **remote** command must agree with the value as coded on the particular event within the subsystem, or that event within the subsystem will not be remotely logged.

**Example:**

```
remote subsystem TKR all log_user log_info
```

In the above example, all messages in subsystem TKR (“all” includes any messages coded for “error,” “info,” or “trace”) will be logged remotely to files specified by log_user and log_info at the remote host.

Use the **list event** and **list active** commands to verify what you set with the **remote** and **noremote** commands.

## 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:**

**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:**

**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:**

**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 monitoring 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:**

**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.

**Syntax:**

**set** pin . . .
remote-logging . . .
timestamp . . .
trace . . .

**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.)

**remote-logging**
Use the **set remote-logging** command to configure remote logging options. When these options are configured from the monitoring environment, the changes take effect immediately, and return to their previously configured settings when the device is rebooted.

**Syntax:**

**set remote-logging** on
off
facility . . .
level . . .
local_id
remote_ip_addr . . .
source_ip_addr ...

**on** Turns remote logging on. Remote logging is now enabled to allow any messages selected by the **remote** command to be actively logged.

**off** Turns remote logging off. All messages selected by the **remote** command will be prevented from being logged.

**facility**

Specifies a value that, in combination with the *level* value, is used by the syslog daemon in the remote workstation to determine where to log messages. This value is used for all remotely-logged ELS messages unless you specify a different value for a particular ELS event, range, group, or subsystem with the **remote** command.

These are all possible syslog facility values:

log_auth
log_authpriv
log_cron
log_daemon
log_kern
log_lpr
log_mail
log_news
log_syslog
log_user
log_uucp
log_local0-7

**level** Specifies a value that, in conjunction with the *facility* value, is used by the syslog daemon in the remote workstation to determine where to log messages. This value is used for all remotely-logged ELS messages unless you specify a different value for a particular ELS event, range, group, or subsystem with the **remote** command.

These are all possible syslog level values:

log_emerg
log_alert
log_crit
log_err
log_warning
log_notice

log_info

log_debug

**local_id**

Specifies a 1-32 character identifier that appears in the remote logging message that you can use to identify which machine logged a particular message.

**remote_ip_addr**

This is an IP address of the remote host where the log files reside.

**source_ip_addr**

Specifies the IP address of the machine that originated the message that is being remotely-logged.

You should use an IP address that is configured in the 2216 for easier identification when the IP address or the hostname is shown in the remotely-logged ELS message. You should also verify that this IP address is quickly resolved to a hostname by the name server, or at least that the name server responds quickly with "address not found."

To determine that the IP address resolves properly enter the **host** command on your workstation as shown:

```
workstation>host 5.1.1.1
host: address 5.1.1.1 NOT FOUND
workstation>
```

If the response takes more than 1 second, select an IP address that resolves more quickly.

**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:**

**set timestamp** [timeofday or uptime or off]

**trace** Use the **set trace** command to configure tracing options. When tracing options are configured from the monitoring environment, the changes take effect immediately, and return to their previously configured settings when the device is rebooted.

**Syntax:**

**set trace**
- decode . . .
- default-bytes-per-pkt . . .
- disk-shadowing . . .
- max-bytes-per-pkt . . .
- memory-trace-buffer-size . . .
- off
- on
- reset
- stop-event . . .
- wrap-mode . . .

**decode [off or 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 or on] or [delete-file 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 or 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 monitoring.

**Note:** Disk shadowing should be set to OFF whenever the WRITE, TFTP software, RETRIEVE system dump, or COPY software commands are issued.

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 monitoring.

**record-size** *bytes*
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.

**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** *hours*
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.

**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.

**Valid Values:** 0, ≥10,000

**Default Value:** 0

**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.

# 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:

**statistics**

### Example:

```
statistics

Subsys  Vector  Exist   String  Active    Heap

  GW     105     101      3411     0         0
 FLT      20       7       184     0         0
 BRS      50       5       201     0         0
 ARP     150     142      7030     0         0
  IP     100     100      2463     2        20
ICMP      30      21       529     0         0
 TCP      60      57      2420     0         0
 UDP      10       6       179     0         0
 BTP      40      13       695     0         0
 RIP      30      22       474     0         0
OSPF      80      73      2859     0         0
MSPF      40      17       593     0         0
TFTP      35      29       819     0         0
SNMP      30      28       821     0         0
 DVM      30      21       589     0         0
  DN     140     115      5842     0         0
  XN      35      21       780     0         0
 IPX     110     110      4705     0         0
CLNP      80      58      1763     0         0
ESIS      40      24       716     0         0
ISIS      80      58      2422     0         0
DNAV      50      26      1314     0         0
 AP2      80      70      1755     0         0
ZIP2      60      51      1859     0         0
R2MP      50      38      1185     0         0
 VIN      90      79      3159     0         0
 SRT     120      94      5040     0         0
 STP      60      32      1590     0         0
  BR      50      30      1616     0         0
SRLY      30      28      1409     0         0
 ETH      60      47      1098     0         0
  SL      50      35       584     0         0
 TKR      60      45      2031     0         0
 X25      70      53      1909     0         0
FDDI      30      27      1155     0         0
SDLC     100      95      4263     0         0
 FRL     130      97      6068     0         0
 PPP     190     186      6394     0         0
X251      50      16       546     0         0
X252      50      34       996     0         0
X253      50      42      1649     0         0
ISDN      50      43      1994     0         0
IPPN      20       4       132     0         0
 WRS      40      33      1938     0         0
 LNM      70      60      3137     0         0
 LLC     170     168      9840     0         0
 BGP      80      74      2477     0         0
 MCF      15       9       244     0         0
 DLS     500     497     24340     0         0
 V25B     30      28      1058     0         0
 BAN      30      29      1223     0         0
 COMP     80      26      1050     0         0
 NBS     100      50      3029     0         0
 ATM     300     216     10808     0         0
 LEC     200     174      7258     0         0
APPN     100      28       467     0         0
ILMI     150      23       487     0         0
```

```
SAAL      30      26       621     0         0
 SVC      30      26       465     0         0
 LES     400     361     22333     0         0
LECS     150     145      5666     0         0

EVLOG      1       1       105     0          0
 NOT      25      15       508     0          0
NHRP     250     211      8193     0          0
 XTP      64      58      2271     0          0
 ESC     150      67      3122     0          0
 LCS      40      22       858     0          0
 LSA      70      61      3506     0          0
 MPC     130      30      1677     3         44
 SCSP     40      34      1234     0          0
ALLC      50      36      1842     0          0
 NDR      50      38      1150     0          0
 MLP     100      93      4006     0          0
 SEC      50      30       688     0          0
ENCR     100       4       194     0          0
  PM      25       6       120     0          0
 DGW      20       9       238     0          0
QLLC      55      54      2411     0          0

Total   6490    4942    215805     5         64

Maximum:7976 vector, 155 subsystem
Memory:71784/620 vector+ 81256/217714 data+ 64 heap=371438Subsys
```

**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 monitoring. This command provides function that is similar to the **packet trace** command described in “Packet-trace Monitoring Commands” on page 175.

**Syntax:**

**trace** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Causes the specified trace event (*subsystem.event#*) to be displayed on the system monitoring.

**group** *groupname*
Allows trace events that were previously added to the specified group to be displayed on the router monitoring.

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_numbe*r is the number of the last event of the specified event range.

Causes the trace events in the specified range for the specified subsystem to be displayed on the system monitoring.

**Example:**

```
trace range gw 19 22
```

Causes the trace events gw.19, gw.20, gw.21, and gw.22 to be displayed on the system monitoring.

**subsystem** *subsystemname*
Allows trace events associated with the specified subsystem to be displayed on the router monitoring.

# 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** event . . .
group . . .
range . . .
subsystem . . .

**event** *subsystem.event#*
Causes the specified message (*subsystem.event#*) to be sent to a network management workstation in an SNMP trap.

**group** *groupname*
Allows messages that were previously added to the specified group to be sent to a network management workstation in an SNMP trap.

**range** *subsystemname first_event_number last_event_number*

Where *first_event_number* is the number of the first event in the specified event range, and *last_event_number* is the number of the last event of the specified event range.

Causes the messages that are in the specified range for the specified subsystem to be sent to a network management workstation in an SNMP trap.

**Example:**

```
trap range gw 19 22
```

Causes the messages in events gw.19, gw.20, gw.21, and gw.22 to be sent to a network management workstation in an SNMP trap.

**subsystem** *subsystemname*
Allows messages associated with the specified subsystem to be sent to a management station in an SNMP trap.

**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.

## View

Use the **view** command to view traced packets.

**Syntax:**

**view** current
first
jump
last
next
prev
search ...

**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.

## Packet-trace Monitoring Commands

This section describes the Packet-trace Monitoring commands. After accessing the Packet-trace Monitoring environment, you can enter Packet-trace Monitoring commands at the `ELS Packet Trace>` prompt.

*Table 17. Packet Trace Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Off | Disables packet tracing. |
| On | Enables packet tracing. Prompts for memory trace buffer size if not previously set. |
| Reset | Clears the trace buffer and resets all associated counters. |
| Set | Configures tracing options. |
| Subsystems | Activates tracing for the ATM subsystems, or displays a summary. |

*Table 17. Packet Trace Monitoring Command Summary (continued)*

| Command | Function |
|---|---|
| Trace-status | Displays information on the status of ATM packet tracing, including configuration and run-time. |
| View | Provides View Captured Packet Trace Buffers Console |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Off

Use the **off** command to disable packet tracing.

**Syntax:**

**off**

## On

Use the **on** command to enable packet tracing.

**Syntax:**

**on**

## Reset

Use the **reset** command to clear the trace buffer and reset all associated counters.

**Syntax:**

**reset**

## Set

Use the **set** command to configure tracing options.

**Syntax:**

**set**

decode
default-bytes-per-pkt
disk-shadowing
max-bytes-per-pkt
memory-trace-buffer-size
stop-event
wrap-mode
exit

For an explanation of the `set` command, see “Set” on page 167.

## Subsystems

Use the **subsystems** command to activate tracing for the ATMsubsystems or display a summary.

**Syntax:**

**subsystems** atm
lec
summary

**Example:**

```
subsystems atm
Network number? 0
ATM Interface is selected
on | off | list [list]? on
Note that SVC  uses VPI = 0, VCI = 5
and ILMI uses VPI = 0, VCI = 16
Beginning of VPI range [0]?
End of VPI range [0]?
Beginning of VCI range [0]? 16
End of VCI range [0]? 16
Tracing event ATM.88: ATM frames
```

**Example:**

```
subsystems lec
Network number? 1
ATM Emulated LAN is selected
on | off | list [list]? on
Trace which types of frames (data, control, both) [both]?
Tracing event LEC.11: data frames over ATM Forum LEC: interface 1
Tracing event LEC.12: control frames over ATM Forum LEC: interface 1
Note that if the user DISABLEs and TESTs this LEC interface,
the LEC trace settings from Talk 6 Config will take effect.
```

**Example:**

```
subsystems summary
Subsystems Being Traced

ATM        net number = 0, VPI Range:     0 -     0
                           VCI Range:    16 -    16
LEC        net number = 1
```

## Trace-Status

Use the **trace-status** command to get updated information regarding packet trace.

**Syntax:**

**trace-status**

**Example:**

```
trace-status
------------------------ Configuration -----------------------------
Trace Status:OFF  Wrap Mode:OFF  Decode Packets:OFF  HD Shadowing:OFF
RAM Trace Buffer Size:0  Maximum Trace Buffer File Size:10000000
Max Packet Bytes Trace:256  Default Packet Bytes Traced:100
Trace File Record Size:2048  Stop Trace Event: None
Maximum Hours to HD Shadow:  24
----------------------- Run-time Status ---------------------------
Packets in RAM Trace Buffer:0   Free Trace Buffer Memory:0
Trace Errors:0  First Packet:0  Last Packet:0
Trace Records Stored on HD:0  Trace Buffer File Size:0
HD-Shadowing Time Exceeded? NO
Has Stop Trace Event Occurred? NO
```

## View

Use the **view** command to enter the View Captured Packet Trace Buffers Monitoring.

For an explanation of the **view** commands, see “View” on page 175.

**Syntax:**

**view** current
first
jump
last
next
prev
search *hexstring*
exit

# ELS Net Filter Monitoring Commands

This section describes explains the commands to manipulate ELS net filters. To enter the filter environment, enter the **filter net** command at the ELS> prompt. Enter the monitoring commands at the `ELS Filter net>` prompt.

*Table 18. ELS Net Filter Monitoring Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Create | Creates a filter and assigns it a number. A maximum of 64 filters is allowed. |
| Delete | Deletes a specified filter number or all filters. |
| Disable | Disables a specified filter number or all filters. |
| Enable | Enables a specified filter number or all filters. |
| List | Lists a specified filter number or all filters. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Create

Use the **create** command to create an ELS net filter.

**Syntax:**

**create** *queue* event *event_name net#_start net#_end*
range *event_range net#_start net#_end*
subsystem *subsystem_name net#_start net#_end*

**queue** The queue for which you are setting the filter. The valid queues are:
Display
Trace
Trap
Remote

**event** *event_name net#_start net#_end*
Specifies the event and net numbers that you are filtering.

If you specify *net#_start* and *net#_end* as the same number, you are filtering on a single net number.

The command **create trap event GW.009 2 10** filters traps for message GW.009 for net numbers 2 through 10.

**range** *event_range net#_start net#_end*
Specifies the range of ELS messages and net numbers that you are filtering.

If you specify *net#_start* and *net#_end* as the same number, you are filtering on a single net number.

The command **create remote range ipx 19 22 3 6** filters all ipx messages beginning with IPX.019 and ending with IPX.022 for net numbers 3 through 6 for remote logging.

**subsystem** *subsystem_name net#_start net#_end*
Specifies the subsystem and net numbers that you are filtering.

If you specify *net#_start* and *net#_end* as the same number, you are filtering on a single net number.

The command **create display subsys ip 1 1**, filters all ELS messages for the ip subsystem that contain net number 1 to the display. All other ip subsystem messages are discarded.

## Delete

Use the **delete** command to delete a specific ELS filter or all ELS filters.

**Syntax:**

**delete** all
filter *filter#*

**all** Deletes all currently configured filters.

**filter** *filter#*
Deletes the filter specified by *filter#*. Use the **list** command to obtain the number for the filter you want to delete.

## Disable

Use the **disable** command to disable a specific ELS filter or all ELS filters.

**Syntax:**

**disable** all
filter *filter#*

**all** Disables all currently configured filters.

**filter** *filter#*
Disables the filter specified by *filter#*. Use the **list** command to obtain the number for the filter you want to disable.

## Enable

Use the **enable** command to enable a specific ELS filter or all ELS filters.

**Syntax:**

**enable** all
filter *filter#*

**all** Enable all currently configured filters.

**filter** *filter#*
Enable the filter specified by *filter#*. Use the **list** command to obtain the number for the filter you want to enable.

## List

Use the **list** command to list a specific ELS filter or all ELS filters.

**Syntax:**

**list** all
filter *filter#*

**all** Lists all currently configured filters.

**filter** *filter#*
Lists the filter specified by *filter#*.

# Chapter 14. Configuring and Monitoring Performance

This chapter describes how to use the Performance monitor configuration and operating commands and includes the following sections:



## Accessing the Performance Configuration Environment

Use the following procedure to access the Performance monitor configuration process.

1. At the OPCON prompt, enter **talk 6**. (For more detail on this command, refer to *The OPCON Process and Commands* in the Software User's Guide.) For example:

   ```
   * talk 6
   Config>
   ```

   After you enter the **talk 6** command, the CONFIG prompt (`Config>`) displays on the terminal. If the prompt does not appear when you first enter configuration, press **Return** again.
2. At the CONFIG prompt, enter the **perf** command to get to the `PERF Config>` prompt.

## Performance Configuration Commands

To configure Performance, enter the commands at the `PERF Config>` prompt.

*Table 19. PERF Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Disable | Disables the collection of CPU utilization statistics or `Talk 2` ELS monitor output. |
| Enable | Enables the collection of CPU utilization statistics or `Talk 2` ELS monitor output. |
| List | Lists the configuration. |
| Set | Sets the reporting period. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Disable

Use the **disable** command to disable collection of CPU utilization statistics and disable the `talk 2` ELS monitor output.

**Syntax:**

**disable** cpu statistics

t2 output

## Enable

Use the **enable** command to enable collection of CPU utilization statistics and enable the `talk 2` ELS monitor output.

**Syntax:**

**enable** cpu statistics

t2 output

## List

Use the **list** command to display the performance monitor configuration.

**Syntax:**

**list**

## Set

Use the **set** command to set the reporting period.

**Syntax:**

**set** *time*

**time** Specifies the short window time.

**Valid Values:** 2 - 30 seconds

**Default Value:** 2

# Accessing the Performance Monitoring Environment

Use the following procedure to access the Performance monitoring commands. This process gives you access to the Performance *monitoring* process.

1. At the OPCON prompt, enter **talk 5**. (For more detail on this command, refer to *The OPCON Process and Commands* in the Software User's Guide.) For example:

   ```
   * talk 5
   +
   ```

   After you enter the **talk 5** command, the GWCON prompt (+) displays on the terminal. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the + prompt, enter the **perf** command to get you to the `PERF Console>` prompt.

   **Example:**

   ```
   + perf
   PERF Console>
   ```

# Performance Monitoring Commands

This section describes the Performance monitoring commands.

*Table 20. PERF Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Clear | Clear the CPU utilization high water statistics and resets the reporting period to a new cycle. |
| Disable | Disables the collection of CPU utilization statistics or `Talk 2` ELS monitor output. |
| Enable | Enables the collection of CPU utilization statistics or `Talk 2` ELS monitor output. |
| List | Lists the configuration. |
| Report | Displays a report of performance statistics. |
| Set | Sets the reporting period. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Disable

Use the **disable** command to disable collection of CPU utilization statistics and disable the `talk 2` ELS monitor output.

**Syntax:**

**disable** cpu statistics

t2 output

## Enable

Use the **enable** command to enable collection of CPU utilization statistics and enable the `talk 2` ELS monitor output.

**Syntax:**

**enable** cpu statistics

t2 output

## List

Use the **list** command to display the performance monitor configuration.

**Syntax:**

**list**

## Report

Use the **report** command to display performance monitor statistics.

**Syntax:**

**report**

**Example:**

```
PERF Console>report
------------------------------------------------------------
KEY:  SW = Short Window = 9 seconds
KEY:  LW = Long Window =  9.0 minutes  (60 x SW)

  CPU UTIL :   Most recent SW                    = 38%
               Most recent LW                    = 33%
               Highest for all SW's              = 92%
               Highest for all LW's              = 52%
               % of time cpu util (SW) was > 60%  = 16%
               % of time cpu util (SW) was > 70%  = 15%
               % of time cpu util (SW) was > 80%  = 1%
               % of time cpu util (SW) was > 90%  = 0%
               % of time cpu util (SW) was > 95%  = 0%
------------------------------------------------------------
```

# Set

Use the **set** command to set the reporting period.

**Syntax:**

**set** *time*

**time** Specifies the short window time.

**Valid Values:** 2 - 30 seconds

**Default Value:** 2

# Part 3. Understanding, Configuring and Operating Interfaces

# Chapter 15. 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"
- "Defining Spare Interfaces" on page 188

## 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 107.)

## 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 Operating Processes" on page 15 , "Accessing the Network Interface Configuration Process" on page 15, and "Accessing the Network Interface Console Process" on page 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" on page 3 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.

## Defining Spare Interfaces

There may be occasions when you will need to define interfaces on your device that do not currently exist. You accomplish this ***dynamic reconfiguration*** of a device by defining spare interfaces while you are configuring the device and then using the console process to activate the interfaces when they are present. See “Configuring Spare Interfaces” on page 60 and “Activate” on page 100 for details.

# Chapter 16. Configuring IEEE 802.5 Token-Ring Network Interfaces

This chapter describes Token-Ring interfaces configuration and operational commands. It includes the following sections:



## Accessing the Token-Ring 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 describes the Token-Ring configuration commands. Enter the commands at the `TKR config>` prompt. Table 21 lists Token-Ring configuration commands.

*Table 21. Token-Ring Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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 | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## 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:**

**list**

**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
```

**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.

# LLC

Use the **LLC** command to access the LLC configuration environment. See “LLC Configuration Commands” on page 223 for an explanation of each of these commands.

**Syntax:**

**llc**

**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:**

**media** *media-type*

**Example:**

```
media unshielded
```

## Packet-Size

Use the **packet-size** command to change maximum packet-size for all Token-Ring networks. Enter the **packet-size** command followed by the desired number of bytes.

**Syntax:**

**packet-size** *bytes*

*Table 22. Token-Ring 4/16 Valid Packet Sizes*

| Network Data Speed | Values (# of bytes) |
|---|---|
| 4 Mbps | 516 to 4498<br>**Note:** If a value greater than 4498 is defined for a 4 Mb TR then the software will set it to 4498. If the user does not specify a value, then the default is 2052. |
| 16 Mbps | 516 to 18144<br>**Note:** If you do not specify a value, then the default is 2052. |

**Note:** If packet sizes are increased, buffer memory requirements will also increase.

## 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.

**Valid values:** Any 12-digit hexadecimal address.

**Default value:** burned-in address (indicated by all zeroes).

**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:**

| | |
|---|---|
| **source-routing** | enable |
| | disable |

## 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:**

| | |
|---|---|
| **speed** | *speed-value* |
| **Example:** | **speed 16** |

# Accessing the Interface Monitoring 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 “Accessing the Network Interface Configuration Process” on page 15 to access the interface monitoring process for the interface described in this chapter. Once you have accessed the desired interface monitoring process, you can begin entering monitoring commands.

## Token-Ring Interface Monitoring Commands

This section summarizes the Token-Ring monitoring commands. Enter commands at the TKR> monitoring prompt. Table 23 lists the monitoring commands.

*Table 23. Token-Ring Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Dump | Displays a dump of the RIF cache. |
| LLC | Displays the LLC monitoring prompt. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**dump**

**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 227 for an explanation of each of these commands.

**Syntax:**

**llc**

# Token-Ring Interfaces and the GWCON Interface Command

While Token-Ring interfaces have their own monitoring 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:

**input overflows**
Specifies the number of frames that were received that were larger than the input buffer size. Frames that are too large to fit into a single input buffer are discarded.

**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 *Event Logging System Messages Guide*

**# 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 17. Using Fast Token-Ring Network Interfaces

This chapter describes how to set software configurable information for fast token-ring interfaces in the router. It includes the following sections:

- "About Fast Token-Ring"
- "Configuring Fast Token-Ring"

## About Fast Token-Ring

Fast Token-Ring FasTR uses the existing IBM 2216 ATM adapters as fast token-ring adapters. It supports IP routing, DLSw, APPN, and SRB (Source Route Bridging).

## Configuring Fast Token-Ring

See "Chapter 1. Getting Started" on page 3 for information about accessing the configuration process. The configuration and operational commands are found in "Chapter 18. Configuring and Monitoring the Fast Token-Ring Network" on page 201.

# Chapter 18. Configuring and Monitoring the Fast Token-Ring Network

This chapter describes FasTR network configuration and operational commands. It includes the following sections:



## Accessing the FasTR Interface Configuration Process

To display the `FasTR config>` prompt, enter the network command followed by the interface number of the FasTR interface. For example:

```
Config>network 0
Fast Token-Ring interface configuration
FasTR 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.

## FasTR Configuration Commands

This section describes the FasTR configuration commands. Enter the commands at the `FasTR config>` prompt. Table 24 lists FasTR configuration commands.

*Table 24. FasTR Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| List | Displays the selected FasTR interface configuration. |
| LLC | Accesses the LLC configuration environment and subcommands. |
| Media | Defaults to fiber. No input allowed. |
| Packet-size | Sets maximum packet-size for FasTR 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 | Defaults to 155 Mbps. No input allowed. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## List

Use the **list** command to display the current configuration for the FasTR interface.

**Note:** If the MAC address is 0, the default station address is used.

**Syntax:**

list

**Example:**

```
list

Fast Token-Ring configuration:

    Packet size (INFO field): 2052
Speed:                    155Mbps
Media:                    Fiber

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

| | |
|---|---|
| Packet size | Maximum FasTR packet size. |
| Speed | Speed of the network, 155Mbps. |
| Media | Type of media the network uses, fiber. |
| 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. |

## LLC

Use the **LLC** command to access the LLC configuration environment. See “LLC Configuration Commands” on page 223 for an explanation of each of these commands.

**Syntax:**

llc

**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

The default media type is fiber. No input allowed.

## Packet-Size

Use the **packet-size** command to set the maximum packet-size for FasTR networks. Enter the **packet-size** command followed by the desired number of bytes.

**Note:** If the packet size is 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 FasTR 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 FasTR 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.

**Valid values:** Any 12-digit hexadecimal address.

**Default value:** burned-in address (indicated by all zeroes).

**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 protocol 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 packets through. Source routing must also be enabled so LLC test response messages can be returned.

**Syntax: source-routing**
enable
disable

**Example: `source-routing enable`**

## Speed

The default speed is 155 Mbps. No input allowed.

# Accessing the Interface Monitoring Process

To display the token-ring monitoring prompt (TKR>), enter the network command followed by the interface number of the FasTR 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 15. Getting Started with Network Interfaces” on page 187 to access the interface monitoring process for the interface described in this chapter. Once you have accessed the desired interface monitoring process, you can begin entering monitoring commands.

# FasTR Interface Monitoring Commands

This section describes the FasTR monitoring commands. Enter commands at the TKR> monitoring prompt. Table 25 lists the monitoring commands.

*Table 25. FasTR Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Dump | Displays a dump of the RIF cache. |
| LLC | Displays the LLC monitoring prompt. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Dump

When source routing is enabled in the FasTR config> process, you can use the **dump** command to request a dump of the RIF cache contents.

**Syntax:**
dump

**Example:**

```
dump

MAC address    State     Usage   RIF
0000C90B1A57   ON_RING   Yes     0220
```

MAC address — Displays the MAC address of the FasTR 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 FasTR 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 227 for an explanation of each of these commands.

**Syntax:**

llc

**Example:**

```
llc

LLC user monitoring
LLC>
```

# FasTR Interfaces and the GWCON Interface Command

While FasTR interfaces have their own monitoring 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 FasTR Interfaces

The following statistics display when you enter the **interface *<net #>*** command for a FasTR interface from the GWCON environment.

```
+i 0
                                    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 Fast Token Ring interface

  Physical address        000000019100
  Network speed           155 Mbps
  Max packet size (INFO)  2052
  Handler state           Ring open

  Hdr Thresh:          0  Bad CRC:             0
  Bad Length:          0  Max Len Exc.:        0
  Rcv Timeout:         0  Fwd Aborts:          0
  Nonzero CPI:         0

  Cells Rcvd:          0  NUD Rcvd:            0
  NUD Bad CRC:         0  Bad HEC:             0

LCD Events:
  AAL0 No Buf:         0  AAL5 No Buf:         0
  NUD No Buf:          0
  Rx No Sysbuf:        0  Tx No Chrmbf:        0

GPDMA Events:
  Tx DMA Error:        0  Rx DMA Error:        0
  Buf Overflow:        0  Virt Mem Res:        0
  Lost Events:         0  Ill. Events:         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″ |
| Slot-Port | Slot number and Port number |
| Self-Test: Pass | Number of times self-test succeeded |
| Self-Test: Fail | Number of times self-test failed |
| Maintenance Failed | Number of maintenance failures |

The following section describes the statistics displayed that are specific to the FasTR interfaces:

| | |
|---|---|
| Physical address | Specifies the physical address of the FasTR interface. |
| Network speed | Specifies the speed of the FasTR 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 FasTR handler. The Handler state counter displays the state of the handler after the self-test runs. |
| Hdr Thresh | Packet header thresholds exceeded. |
| Bad CRC | Packets received with bad CRC. |
| Bad Length | Packets received with bad length. |
| Max Len Exc. | Packets received exceeding maximum length. |
| Rcv Timeout | Timeouts on received packet reassembly. |
| Fwd Aborts | Received packets terminated with a forward abort. |
| Nonzero CPI | Packets received with CPI field not set to zero. |
| Cells Received | Cells received (not packets). |

| | |
|---|---|
| NUD Rcvd | Non-user data fields received. |
| NUD Bad CRC | Non-user data fields received with bad CRC_10. |
| Bad HEC | Cells received with bad Header Error Check. |
| LCD Events | |
| AAL0 No Buf | AAL0 cells dropped due to lack of pools buffers. |
| AAL5 No Buf | AAL5 cells dropped due to lack of pools buffers. |
| NUD No Buf | Non-user data dropped due to lack of pools buffers. |
| Rx No Sysbuf | Packets received but dropped because no system buffers were available. |
| Tx No Chrmbuf | Transmit packets dropped because no adapter buffers were available. |
| GPDMA Events | |
| Tx DMA Error | Transmit DMAs with errors. |
| Rx DMA Error | Receive DMAs with errors. |
| Buf Overflow | Received packets that exceeded the real buffer size. |
| Virt Mem Res | Virtual Memory resource events, writing cells into virtual memory. |
| Lost Events | Events lost because the receive queue was full. |
| Ill. Events | Unrecognized events. |

# Chapter 19. Using FDDI

This chapter describes how to set software-configurable information for the Fiber Distributed Data Interface (FDDI) in the router.

This chapter contains the following sections:

- "Fiber Distributed Data Interface (FDDI) Overview"

## Fiber Distributed Data Interface (FDDI) Overview

Fiber Distributed Data Interface (FDDI) is described by the ANSI X3T9.5 and ISO 9314 committees as a dual counter-rotating ring that operates at a defined speed of 100 Mbps.

In many ways, FDDI is similar to the IEEE 802.5 token-ring, although there are differences, some of which are described in "Differences Between FDDI and Token-Ring" on page 210.

### Token-Passing Ring Network

FDDI is defined as a token-passing protocol. Each station has the chance to transmit data when a token passes. A station can decide how many frames it will transmit using an algorithm that permits "bandwidth" allocating.

FDDI also allows a station to transmit many frames without releasing the token in a way that is similar to the IEEE 802.5 token-ring standard.

An FDDI ring network consists of a set of stations/devices connected as a serial string of stations/devices and transmission media to form a physically closed loop. Information is transmitted sequentially as a stream of suitably encoded signals from one active station/device to the next active one.

Each station/device generally regenerates and repeats each token and can serve as the means of attaching one or more stations/devices to the network.

### Primary and Secondary Rings

FDDI defines two rings:

- The *primary ring*, which is similar to the main ring path in a token-ring network.
- The *secondary ring*, which is similar to the backup ring path in a token-ring network.

Each ring path consists of two fibers, each fiber transmitting one signal; one is pushed and one is pulled in a device. Each fiber is equivalent to a pair of copper conductors. The physical approach in terms of fiber optics is similar to physical fiber optic token-ring paths.

### Attachment of Devices

FDDI permits many attachment units:

- Stations or devices

- Concentrators
- Bridges

These units can be attached to FDDI networks in various ways, similar to those for token-ring networks.

## Differences Between FDDI and Token-Ring

The main differences between FDDI and token-ring techniques are:

- A device can be attached directly to rings without a concentrator, such as a multi-station access unit (MSAU) on a token ring.
- A device can be attached to either or both of the primary and secondary rings.

FDDI defines two device classes, A and B, to differentiate between devices that attach to one ring or both rings, as described in the next section.

## Device Classes A and B

FDDI defines two device classes:

- A **Class A** device attaches to both rings directly.

  It can be a station, called a *Class A station* or *Dual Access Station (DAS)*, or it can be a Concentrator, called a *Dual Access Concentrator (DAC)*
- A **Class B** device attaches to only one of the rings directly or through a concentrator.

  It can be a station, called a *Class B station* or *Single Access Station (SAS)*, or it can be a Concentrator, called a *Single Access Concentrator (SAC)*

# FDDI Network Diagram

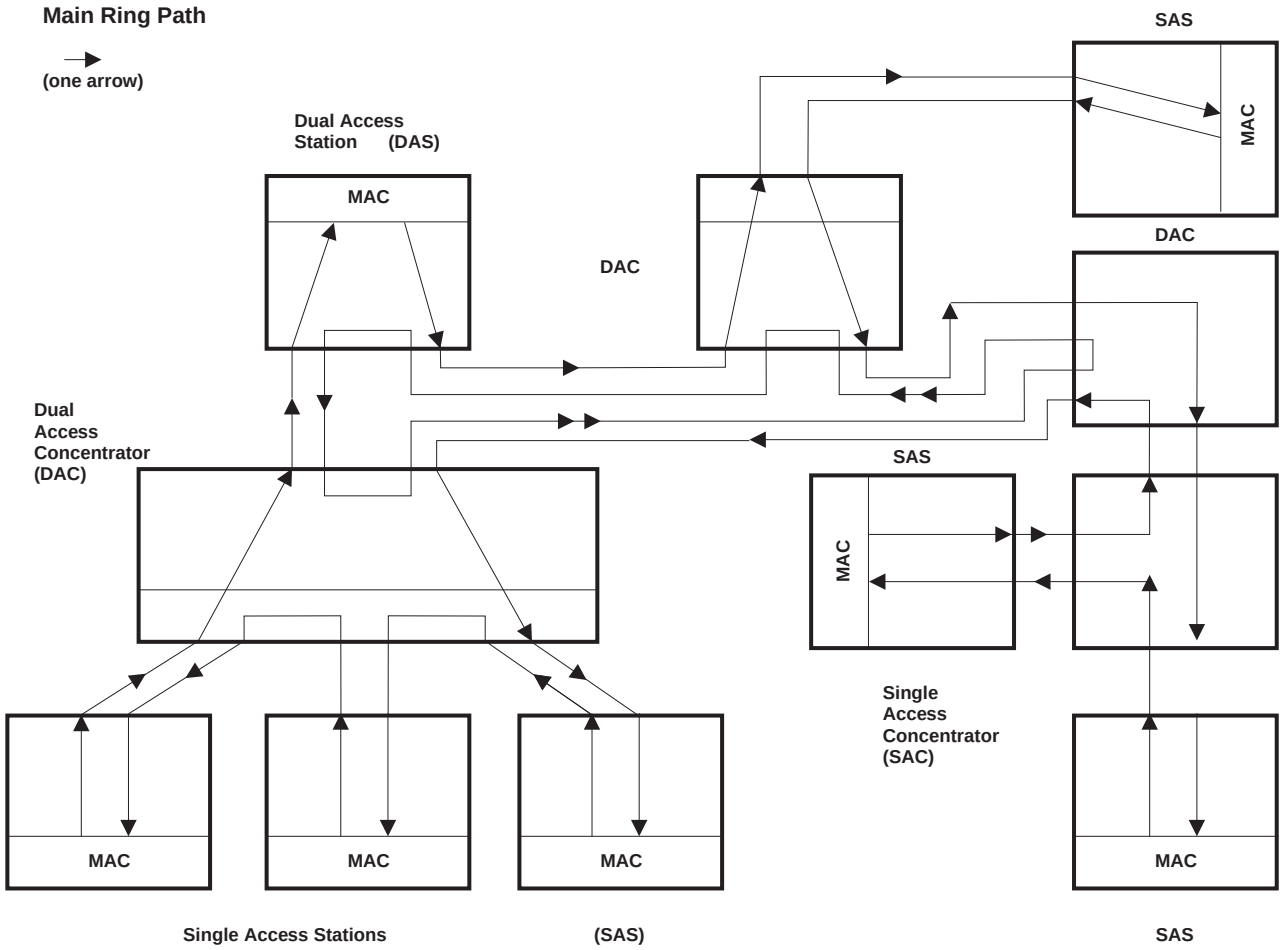



*Figure 15. FDDI Network Diagram. This diagram shows Single Access Stations (SAS), Dual Access Stations (DAS), Single Access Concentrators (SAC), and Dual Access Concentrators (DAC) in one Main Ring Path of an FDDI network.*

# Chapter 20. Configuring and Monitoring FDDI

This chapter describe the FDDI interface configuration and operational commands. It includes the following sections:

- "Accessing the FDDI Configuration Commands"
- "FDDI Configuration Commands"
- "Accessing FDDI Monitoring Commands" on page 216
- "FDDI Monitoring Commands" on page 216

## Accessing the FDDI Configuration Commands

You can access FDDI configuration from Talk 6. To do so, enter the **add device** command to add an FDDI interface to the network and assign an interface number to it, and then use the **network** command to access the FDDI interface as shown in the following example:

```
800 Config>  add device fddi
SK-NET FDDI device in slot 0 port 1 as interface #2
Use "net 2" to configure SK-NET FDDI parameters

800 Config>  network ?
0  :CHARM ATM Adapter
1  :ATM Token Ring LAN Emulation: elan1
2  :SK-NET FDDI

800 Config>  network 2
FDDI Interface Configuration
FDDI Config>
```

This will get you to the `FDDI Config>` prompt.

When you are finished, enter **Exit** to return to the previous prompt level.

## FDDI Configuration Commands

This section describes the FDDI configuration commands as shown in Table 26. Enter the commands from the `FDDI Config>` prompt.

*Table 26. FDDI Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| LLC | Provides access to the LLC configuration environment. |
| List | Displays the selected FDDI configuration. |
| Set | Sets FDDI parameters. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

### LLC

Use the **LLC** command to access the LLC configuration environment. See "LLC Configuration Commands" on page 223 for an explanation of LLC command.

**Syntax:**

**llc**

## List

Use the **list** command to display the current configuration for the FDDI.

**Syntax:**

**list** all
ler
pmf
tmax
tmin
treq
tvx
userdata

**all** Lists all of the output for the various parameters that follow.

**ler** Lists the link error rate alarms and cutoff values for port A and port B.
a -- lists the link error rate alarms and cutoff values for port A.
b -- lists the link error rate alarms and cutoff values for port B.

**pmf** Displays the PMF Password (maximum of 8 characters).

**tmax** Lists the Maximum Token Rotation Time (in milliseconds).

**tmin** Lists the Minimum Token Rotation Time (in milliseconds).

**treq** Lists the Requested Target Token Rotation Time (in milliseconds).

**tvx** Lists the Valid transmission timer expiration (in microseconds)

**userdata**
Displays the user data (maximum of 32 characters).

## Set

Use the **set** command to configure FDDI.

**Syntax:**

**set** ler
pmf
tmax
tmin
treq
tvx
userdata

**ler port# type** Sets the alarm and cutoff values for port A and port B as follows:

**ler a alarm**
Sets the alarm values for port A.

**ler a cutoff**
Sets the cutoff values for port A.

**ler b alarm**
Sets the alarm values for port B.

**ler b cutoff**
Sets the cutoff values for port B.

**Valid values and defaults**

| Alarm or Cutoff | Valid Values | Default |
|---|---|---|
| **Alarm** | 4 to 15 | 8 |
| **Cutoff** | 4 to 15 | 7 |

**pmf**
Sets the PMF Password (maximum of 8 characters).

**tmax**
Sets the Maximum Token Rotation Time (in milliseconds) that this station can accept. Commonly referred to in FDDI specifications as T_Max.

**Valid values:** 5 to 165 milliseconds

**Default:** 165 milliseconds

**tmin**
Sets the Minimum Token Rotation Time (in milliseconds) that this station can accept. If the negotiated TTRT is less than this value, then the adapter will not provide proper service to the layers above it. Commonly referred to in FDDI specifications as T_Min.

**Valid values:** 5 to 165 milliseconds

**Default:** 5 milliseconds

**treq**
Sets the Requested Target Token Rotation Time (in milliseconds) that this station will bid during initialization. Commonly referred to in FDDI specifications as T_Req.

**Valid values:** 5 to 165 milliseconds

**Default:** 165 milliseconds

**tvx**
Sets the Valid transmission timer expiration (in microseconds). This timer is reset every time a valid frame or nonrestricted token is seen by the station. If the timer expires, it indicates that traffic is not circulating properly on the ring and therefore the claim process is started. Commonly referred to in FDDI specifications as TVX.

**Valid values:** 2500 to 10 000 microseconds

**Default:** 2500 microseconds.

**userdata**
Sets the User data (maximum of 32 characters).

## Accessing FDDI Monitoring Commands

You can access FDDI configuration from Talk 5 by entering the **network** command to access the FDDI interface as shown in the following example:

```
800+  network ?
0  :CHARM ATM Adapter
1  :ATM Token Ring LAN Emulation: elan1
2  :SK-NET FDDI

800 +  network 2
FDDI Interface
FDDI>
```

This will get you to the FDDI> prompt.

When you are finished, enter **Exit** to return to the previous prompt level.

## FDDI Monitoring Commands

The monitoring commands for FDDI are:

*Table 27. FDDI Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Displays the FDDI adapter information |
| LLC | Displays the LLC monitoring prompt. |
| Srt-stats | Displays the FDDI bridging statistics. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## LLC

Use the **LLC** command to access the LLC configuration environment. See “LLC Configuration Commands” on page 223 for an explanation of LLC command.

**Syntax:**

**llc**

## List

Use the **list** command to display the current FDDI configuration parameters.

**Syntax:**

**list**

**Example:**

```
 FDDI> list
MAC Address: 00:00:5A:00:00:01
```

## Srt-stats

Use the **srt-stats** command to display the hardware assisted bridged statistics on this interface.

**Syntax:**

**srt-stats**

**Example:**

```
srt-stats
Port Supports:               Transparent Bridging Only

Frames received:                           806
Bytes received:                          34588
Maximum size of filter table in adapter:  4088
Number of entries in filtr table:            0
Number of dynamic entries in filter table:   0
```

## FDDI Interfaces and the GWCON Command

While the FDDI interfaces have their own monitoring processes for monitoring, the router also displays complete statistics for installed network interfaces when you use the interface command from the GWCON environment.

## Statistics Displayed from FDDI Interfaces

The following statistics are displayed when you enter the **interface** *net#* command for a FDDI interface from the GWCON environment.

```
                                                Self-Test   Self-Test  Maintenance
Nt Nt'  Interface  Slot-Port                       Passed     Failed     Failed
0   0    FDDI/0    Slot: 1   Port: 1                 1          0          0

  IEEE 802.2/FDDI MAC/data-link on SK-NET FDDI interface
Address: 00:60:94:C4:00:40
UNA:     00:00:5A:02:2D:1E -> DNA: 00:00:5A:02:2D:1E
------------------------------------------------------------------------
ECM State Machine:        IN
PCM State Machine Port A: SIGNAL
PCM State Machine Port B: ACTIVE
CFM State Machine Port A: ISOLATED
CFM State Machine Port B: CONCATENATED
CF State Machine:         C_WRAP_B
MAC Current Path:         PRIMARY
RMT State Machine:        RING_OP
------------------------------------------------------------------------
TVX expired ct: 0
Beacon ct:      0
Claim ct:       0
RingOp ct:      1
------------------------------------------------------------------------
PHYA:LEM_Ct:    0  LEM_Reject_Ct:    0  LCT fails: 40
     Alarm: 10°-8   Cutoff: 10°-7   Estimate: 10°-15
PHYB:LEM_Ct:    0  LEM_Reject_Ct:    0  LCT fails: 40
     Alarm: 10°-8   Cutoff: 10°-7   Estimate: 10°-15
------------------------------------------------------------------------
T_Notify 10 sec, SMT frames in:55363  SMT frames out:35317
------------------------------------------------------------------------
Frames:211764, Errors:0, Losts:0, Xmts:144058, Copied:171046, Not Copied: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 Passed**
Number of times self-test succeeded.

**Self-Test Failed**
Number of times self-test failed.

**Maintenance Failed**
Number of maintenance failures.

The following section describes the statistics displayed that are specific to the FDDI interfaces:

**Address**
Specifies the physical address of the FDDI interface.

**UNA** Specifies the physical address of the upstream neighbor.

**DNA** Specifies the physical address of the downstream neighbor.

**ECM State Machine**
Entity Coordination Management controls the management of the media interface, including all the ports at the node. It also controls the optical bypass.

```
OUT
IN
TRACE
LEAVE
PATH-TEST
INSERT
CHECK
DEINSERT
```

**PCM State Machine**
Physical Connection Management controls the management of the physical connection between a port being managed and another port in the adjacent node.

```
OFF
BREAK
TRACE
CONNECT
NEXT
SIGNAL
JOIN
VERIFY
ACTIVE
MAINT
```

**CFM State Machine**
Configuration Management manages the configuration of MACs and ports within a node.

```
ISOLATED
LOCAL
SECONDARY
PRIMARY
CONCATENATED
THRU
```

**CF State Machine**
Attachment configuration.

```
ISOLATED
LOCAL_A
LOCAL_B
```

```
LOCAL_AB
LOCAL_S
WRAP_A
WRAP_B
WRAP_AB
WRAP_S
C_WRAP_A
C_WRAP_B
C_WRAP_S
THRU
```

**MAC Current Path**
Current path which this MAC is inserted.

```
ISOLATED
LOCAL
SECONDARY
PRIMARY
```

**RMT State Machine.**
Ring Management controls the timing of the MAC management frames.

```
ISOLATED
NON_OP
RING_OP
DETECT_BEACON
NON_OP_DUP
RING_OP_DUP
DIRECTED
RM-TRACE
DETECT_CLAIM
DETECT_IDLE
```

**TVX expired ct**
Number of times TVX expired.

**Beacon ct**
Number of times beacon state entered.

**Claim ct**
Number of times claim state entered.

**RingOp ct**
Number of times ring has entered operational state.

**LEM_Ct**
Link error monitor error count.

**LCT fails**
Count of consecutive times the link confidence test has failed.

**Alarm** Estimate at which a link connection will generate an alarm.

**Cutoff** Estimate at which a link connection will be broken.

**Estimate**
Long term average link error rate.

**Frames**
Number of frames received.

**Errors** Number of frames detected in error.

**Losts** Number of format errors during reception.

**Xmts** Number of frames transmitted.

**Copied**
Number of frames copied.

**Not Copied**
Number of frames not copied.

**T_Notify**
Neighbor notification timer.

**SMT frames in**
Number of SMT frames received.

**SMT frames out.**
Number of SMT frames sent.

# Chapter 21. Using LLC Interfaces

This chapter describes how to set software configurable information for Logical Link Control (LLC) interfaces in the router.

Logical Link 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 protocol by entering an **LLC** command.

# Chapter 22. Configuring and Monitoring LLC Interfaces

This chapter describes how to configure specific LLC interfaces in the router by using either the interface commands or the GWCON interface command.

Logical Link Level can be thought of as a “sub-protocol”. It is not accessed directly from either the Talk 6 (configuration) or the Talk 5 (monitoring) 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 Configuration Process

Access the configuration commands for the protocol you wish to configure over LLC:

- Token Ring, as described in “Chapter 16. Configuring IEEE 802.5 Token-Ring Network Interfaces” on page 189
- Point-to-Point, as described in “Chapter 41. Using Point-to-Point Protocol Interfaces” on page 511
- Frame Relay, as described in “Chapter 39. Using Frame Relay Interfaces” on page 457
- FDDI, as described in “Chapter 19. Using FDDI” on page 209

Each of these prompt levels 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 Token-Ring Interface Configuration Process” on page 189).

This section summarizes and then explains all of the LLC configuration commands. These commands ( Table 28) enable you to configure LLC when you need to pass packets over a SNA network.

*Table 28. LLC Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Displays the selected LLC configuration. |
| Set | Sets the timers associated with LLC, and the size of the transmit and receive windows. |

*Table 28. LLC Configuration Command Summary (continued)*

| Command | Function |
|---|---|
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

# List

Use the **list** command to display the current configuration for the LLC.

**Syntax:**

**list**

**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:**

**set** n2-max-retry *count*
n3-frames-rcvd-before-ack *count*
nw-acks-to-inc-window *count*
rw-receive-window *count*
t1-reply-timer *seconds*
t2-receive-ack-timer *seconds*
ti-inactivity-timer *seconds*
tw-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]?
```

## Accessing the Interface monitoring Process

Access the monitoring commands for the protocol you wish to monitor over LLC:

- Token Ring, as described in “Chapter 16. Configuring IEEE 802.5 Token-Ring Network Interfaces” on page 189
- Point-to-Point, as described in “Chapter 42. Configuring and Monitoring Point-to-Point Protocol Interfaces” on page 525
- Frame Relay, as described in “Chapter 40. Configuring and Monitoring Frame Relay Interfaces” on page 475
- FDDI, as described in “Chapter 19. Using FDDI” on page 209

Each of these prompt levels has an LLC command. Enter **LLC** to access the LLC monitoring 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 29. LLC Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Clear-Counters

Use the **clear-counters** command to clear all the LLC statistical counters.

**Syntax:**

**clear-counters**

## List

Use the **list** command to display interface, service access point (SAP), and session information.

**Syntax:**

**list** interface
sap . . .
session

**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 monitoring 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 227 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 "Set" on page 225.

**Attention:** Changing LLC parameters from the default can affect how the LLC protocol works.

**Syntax:**

**set** n2-max_retry *count*
n3-frames-rcvd-before-ack *count*
nw-acks-to-inc-ww *count*

t1-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.

**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.

**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.

**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.

**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.

**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.

# Chapter 23. Using the Ethernet Network Interface

This chapter describes how to use the Ethernet interface. It includes the following sections:



## 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
```

These statistics have the following meaning:

**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.

# Chapter 24. Configuring and Monitoring the Ethernet Network Interface

This chapter describes Ethernet interface configuration and operational commands. It includes the following sections:



## Accessing the Ethernet 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 **talk 6**. (For more detail on this command, refer to "Chapter 4. The OPCON Process and Commands" on page 29.) For example:

   ```
   * talk 6
   Config>
   ```

   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.
4. 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.

*Table 30. Ethernet Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Connector-Type | Sets the connector 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. |
| Physical-Address | Sets the physical MAC address. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## 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:**

**connector-type** *name*

## 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*

## 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:**

**list** all

**Example:**

```
list all

Connector type:           AUI (10BASE5)
MAC Address:              12:15:00:FA:00:FE
```

## Physical-Address

Use the **physical-address** command to set the physical (MAC) address.

**Syntax: physical-address** ***address***

**physical-address**

This command lets you indicate whether you want to define a locally administered address for the Ethernet interface's MAC sublayer address, or use the default burned-in address (indicated by all zeros). The MAC sublayer address is the address that the Ethernet interface uses to receive and transmit frames.

**Note:** Pressing **Enter** leaves the value the same. Entering **0** causes the router to use the burned-in address. The default is to use the burned-in address.

**Valid Values:** Any 12-digit hexadecimal address.

**Default Value:** burned-in address (indicated by all zeros).

**Example:**

```
physical-address

MAC address in 00:00:00:00:00:00 form []? 12:15:00:FA:00:FE
```

## Accessing the Ethernet Interface Operating Process

To monitor information related to the Ethernet Network Interface, access the interface monitoring 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 “Configuration” on page 102 for sample output of the **configuration** command.

3. Enter the **network** command and the number of the Ethernet interface. In this example:

   ```
   + network 0
   ETH>
   ```

   The Ethernet monitoring prompt is displayed. You can now view information about the Ethernet interface by entering monitoring commands.

## Ethernet Interface Monitoring Commands

This section summarizes and explains the Ethernet monitoring commands. Enter commands at the ETH> prompt. Table 31 lists the monitoring commands.

*Table 31. Ethernet monitoring command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Collisions | Displays collision statistics for the specified Ethernet interface. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**collisions**

**Example:**

## Ethernet Interface Monitoring Commands (Talk 5)

```
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
```

# Chapter 25. Using the 10/100 Mbps Ethernet Network Interface

This chapter describes how to use the 10/100 Mbps Ethernet interface. It includes the following sections:

- "Displaying 10/100 Mbps Ethernet Statistics"

## Displaying 10/100 Mbps Ethernet Statistics

You can use the **interface** command from the GWCON environment to display the following statistics.

```
+i 0
                                          Self-Test  Self-Test Maintenance

Nt  Nt'  Interface Slot-Port                 Passed     Failed      Failed
0   0    Eth/0     Slot: 1   Port: 1            1          0           0

  Ethernet/IEEE 802.3 MAC/data-link on 100MB Ethernet interface

  Physical address         10005A991431
  PROM address             10005A991431
  Actual address           10005991431
  Adapter Level            0
  Configured Duplex         : Auto-Negotiation
  Actual Duplex             : Half Duplex
  Configured Speed          : Auto-Negotiation
  Actual Speed              : 100 Mbps

  Input statistics:
   failed, packet too long            0   failed, CRC error            0
   failed, alignment error            0   failed, receive Overflow     0
  *receive collision                  0  *missed frame                 0
 **frames filtered                    0   receive underrun             0

   Output statistics:
   one retry                          0   single collision             0
   multiple collisions                0   failed, transmit underflow   0
   failed, excess collisions          0   failed, loss of carrier      0
   late collisions                    0   more than one retry          0
   buffer error                       0   total collisions             0
   excessive deferal                  0   deferred                     0
   memory error                       0

  * cannot be cleared
 ** cleared automatically when read
```

These statistics have the following meaning:

**Nt** Global network number.

**Nt′** This field is for the serial interface card. Disregard the output.

**Interface**
Interface name and its instance 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.

**Actual address**

**Adapter level**

**Configured duplex**
The value configured for duplex. Values can be Half Duplex, Full Duplex, or Auto-Negotiation.

**Actual duplex**
The value at which the adapter is presently operating. It might be different from the value configured, depending on the switch capability. If the adapter is not Up, the value displayed will be ″Unknown″. Otherwise the value can be Half Duplex or Full Duplex.

**Note:** The value indicated here might not be accurate. This is due to the implementation of the negotiation and link signaling support in the manufacturer's products.

**Configured speed**
The value configured for speed. Values can be 10 Mbps, 100Mbps, or Auto-Negotiation.

**Actual speed**
The speed at which the adapter is presently operating. It might be different from the speed configured, depending on the switch capability. If the adapter is not Up, the value displayed will be ″Unknown″. Otherwise the value can be 10 Mbps or 100 Mbps.

**Note:** The value indicated here might not be accurate. This is due to the implementation of the negotiation and link signaling support in the manufacturer's products.

**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, alignment error**
The Failed, Framing Error counter increments when the interface receives a packet where the length in bits is not a multiple of eight.

**failed, receive overflow**
Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into memory buffer before the internal FIFO overflowed.

**receive collision**
Indicates the total number of collisions encountered by the receiver support on the adapter.

**Note:** This counter cannot be cleared by the **clear statistics** command because it is maintained on the adapter. The **test network** command is the only way to reset this counter.

**missed frame**
Indicates the number of incoming receive frames lost due to unavailability of a receive buffer in the system. This error indicates that the system is not processing received frames as fast as they are being received from the local network.

**Note:** This counter cannot be cleared by the **clear statistics** command because it is maintained on the adapter. The **test network** command is the only way to reset this counter.

**frames filtered**
Indicates the number of incoming frames that were discarded by the adapter. This counter is updated only when bridging is enabled.

**Note:** This counter is maintained on the adapter, and is cleared every time it is read. This counter will be cleared by the **interface statistics** and the **test network** commands.

**receive underrun**
Indicates the number of times the adapter did not have a second buffer to store a long frame (requiring more than one buffer).

**Output statistics:**

**one retry**
Indicates that exactly one retry was needed to transmit a frame. 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.

**failed, transmit underflow**
Transmit underrun indicates that transmitter has truncated a message because it could not read data from the memory fast enough. It also indicates that the FIFO on the adapter has emptied out before the end of the frame was reached. IFO into memory buffer before the internal FIFO overflowed.

**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, loss of carrier**
Loss of carrier is set when the carrier is lost during transmission. The adapter does not retry upon loss of carrier. It will continue to transmit the whole frame until done.

**late collisions**
A late collision indicates that a collision has occurred after the first channel slot time has elapsed. The adapter does not retry on late collisions.

**more than one retry**
More than one retry indicates that more than one retry was needed to transmit a frame.

**buffer error**
Buffer error occurs if there is a memory corruption problem in the system, or under certain FIFO underflow conditions on the adapter.

**total collisions**
The Total Collisions counter increments by the number of collisions a packet incurs.

**excessive deferral**
Excessive deferral indicates that the transmitter on the adapter has experienced Excessive Deferral on this a transmit frame, where Excessive Deferral is defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard.

**deferred**
Deferred indicates the number of times the adapter had to defer while trying to transmit a frame. This condition occurs if the DMA channel is busy when the adapter is ready to transmit.

**memory error**
Memory errors occur when the adapter is not given access to the system interface bus within the programmable length of time. This error will normally occur during transmit operations, indicating transmit underrun.

# Chapter 26. Configuring and Monitoring the 10/100 Mbps Ethernet Network Interface

This chapter describes the 10/100 Mbps Ethernet interface configuration and operational commands. It includes the following sections:



## 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 **talk 6**. (For more detail on this command, refer to "Chapter 4. The OPCON Process and Commands" on page 29.) For example:

   ```
   * talk 6
   Config>
   ```

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

   ```
   Config> network 0
   Ethernet 100 interface configuration
   ETH100 Config>
   ```

   The 10/100 Mbps Ethernet configuration prompt (`ETH100 Config>`), is displayed.

## 10/100 Mbps Ethernet Configuration Commands

This section describes the 10/100 Mbps Ethernet configuration commands. Enter the commands at the `ETH config>` prompt.

*Table 32. 10/100 Mbps Ethernet Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Duplex | Sets the duplex mode. |
| 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. |
| Physical-Address | Sets the physical MAC address. |
| Speed | Sets the link speed. |

*Table 32. 10/100 Mbps Ethernet Configuration Command Summary (continued)*

| Command | Function |
|---|---|
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Duplex

Use the **duplex** command to set the duplex mode.

**Syntax:**

**duplex** half_duplex
full_duplex
auto

**Half_duplex**
The interface will not transmit while receiving or receive while transmitting.

**Full_duplex**
The interface will transmit and receive simultaneously.

**Auto** The interface will automatically select half-duplex or full duplex depending on the link partner’s capability.

## 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** for the type.

**Syntax:**

**IP-encapsulation** *type*

**Example:** `IP-encapsulation e`

## List

Use the **list** command to display the current configuration for the 10/100 Mbps Ethernet interface.

**Syntax:**

**list** all

**Example:**

```
list all

The duplex is   HALF DUPLEX
The speed is   100Mb
No IPX interface configuration
IP Encapsulation:   Ether

MAC Address:        023456789A56
```

## Physical-Address

Use the **physical-address** command to set the physical (MAC) address.

**Syntax:**

**physical-address** *address*

**physical-address**

This command lets you indicate whether you want to define a locally administered address for the Ethernet interface's MAC sublayer address, or use the default burned-in address (indicated by all zeros). The MAC sublayer address is the address that the Ethernet interface uses to receive and transmit frames.

**Note:** Pressing **Enter** leaves the value the same. Entering **0** causes the router to use the burned-in address. The default is to use the burned-in address.

**Valid Values:** Any 12-digit hexadecimal address.

**Default Value:** burned-in address (indicated by all zeros).

**Example:**

```
physical-address
MAC address in 00:00:00:00:00:00 form []? 12:15:00:FA:00:FE
```

## Speed

Use the **speed** command to set the speed used by this interface.

**Syntax:**

**speed** ten
hundred
auto

**Ten** The interface will operate at 10 Mbps.

**Hundred**
The interface will operate at 100 Mbps

**Auto** The interface will automatically select the speed (10 Mbps or 100 Mbps) depending on the link partner's capability.

# Accessing the 10/100 Mbps Interface Monitoring Process

To monitor information related to the 10/100 Mbps Ethernet Network Interface, access the interface monitoring 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 "Configuration" on page 102 for sample output of the **configuration** command.
3. Enter the **network** command and the number of the Ethernet interface. In this example:

```
+ network 0
ETH100>
```

The 10/100 Mbps Ethernet monitoring prompt is displayed. You can now view information about the 10/100 Mbps Ethernet interface by entering monitoring commands.

## 10/100 Mbps Ethernet Interface Monitoring Commands

This section summarizes the 10/100 Mbps Ethernet monitoring commands. Enter commands at the ETH100> prompt. Table 33 lists the monitoring commands.

*Table 33. Ethernet Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Collisions | Displays collision statistics for the specified Ethernet interface. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**collisions**

**Example:**

```
Eth100> 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
```

# Chapter 27. 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 16.

## Simple LAN Emulation Network

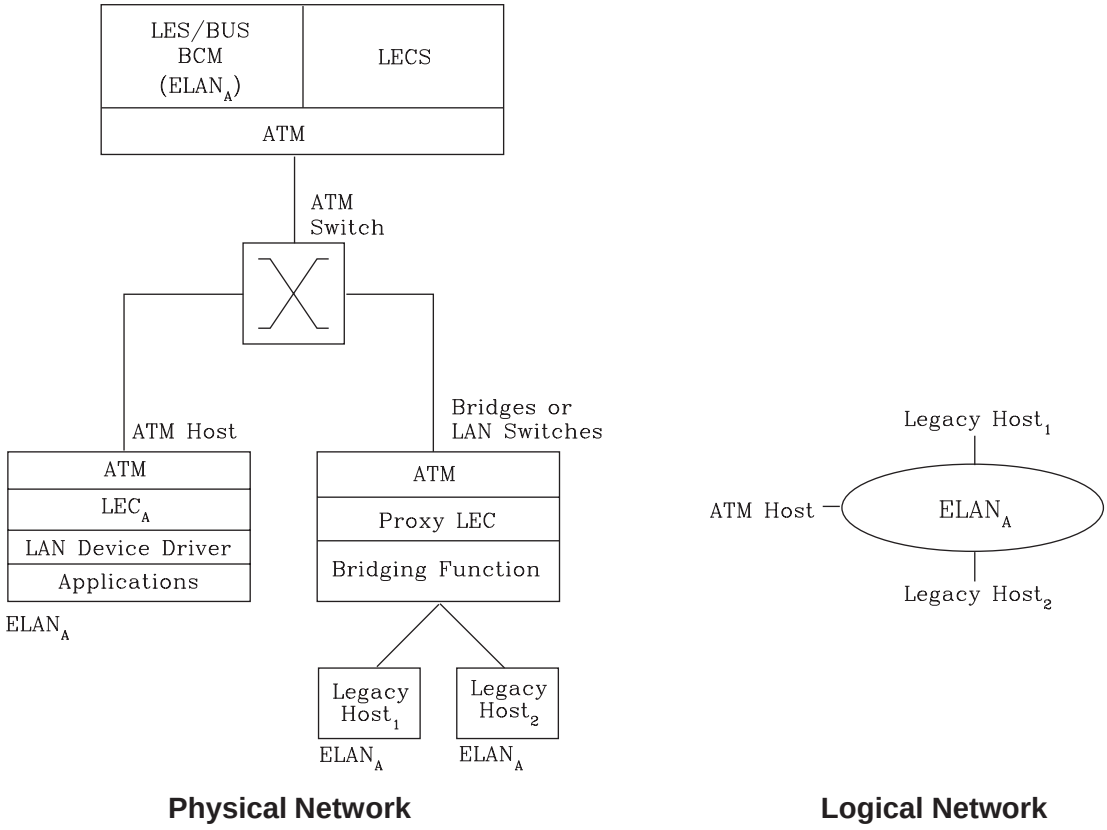



*Figure 16. Physical and Logical Views of a Simple LAN Emulation Network*

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 (LECs)**
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.

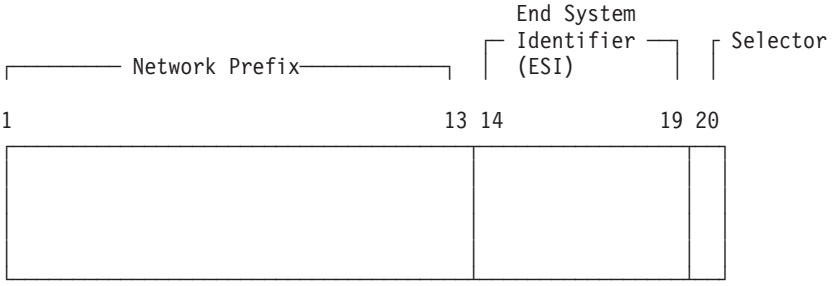
**Broadcast and Unknown Server (BUS)**
A LAN emulation service component responsible for the delivery of multicast and unknown unicast frames.

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 hierarchical addressing:




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 an end system activates, it attempts to register its ESIs with its ATM switch using the Interim Local Management Interface (ILMI).

The 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 to 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. You can use the MAC address as an ESI for some or all of the router's ATM addresses. Alternatively, you can 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. You can 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 addresses. Additional ESIs can be configured on each ATM interface.

Each ESI can 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 require you to explicitly configure a selector from the configured selector range. LES/BUSs are an example of such a component. Other components, such as Classical IP clients, allow the selector to be automatically assigned at run-time. You do not have to choose the selector because the router does this when it activates. This selector is not guaranteed to be consistent across router restarts. Automatic selector assignment is useful only for those ATM components whose ATM address does not have to be already known 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 253
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 253, 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 a 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 correct LES for that LE client. The LECS identifies an 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). An 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 an 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 1 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 1 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 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) means 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) means 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 (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 can be assigned to LESs based upon a MAC address or a route descriptor. Because a LAN destination uniquely identifies an 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 correct 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 by configuring these policy values:

  ```
  (Accounting,  LES_A)
  (Engineering, LES_A)
  ```

  or to different ELANs by configuring these 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, you 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 three 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 17; 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 261.

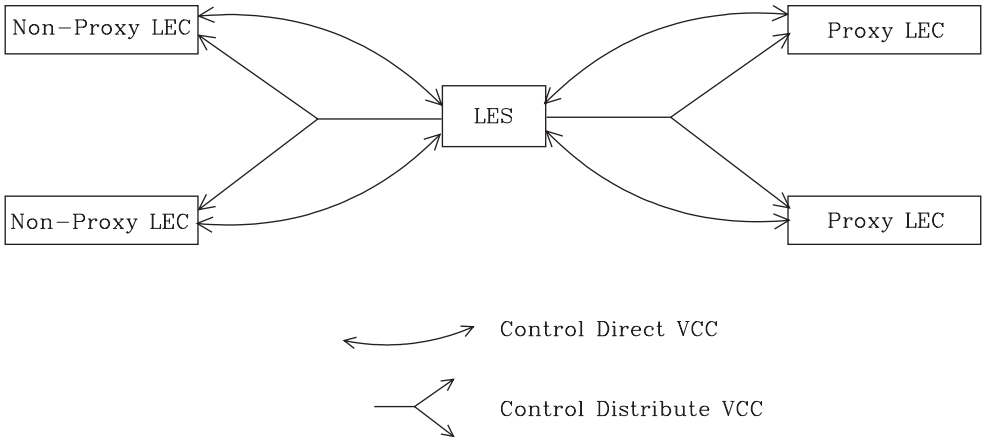



*Figure 17. Default Connections Between LE Clients and the LES*

The following ATM connections are established between the LE client and the LES:

**Control Direct VCC (bidirectional 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 18 on page 262 shows partitioned Multicast Forward VCCs.

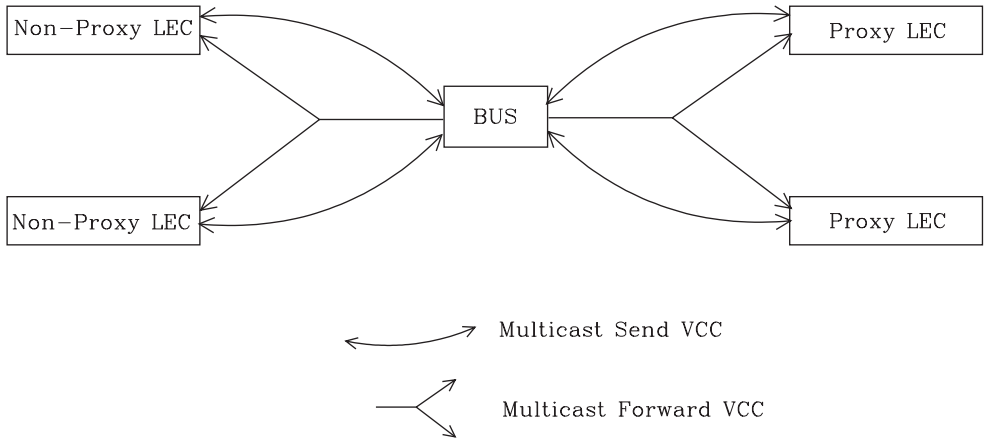



*Figure 18. Default Connection Between LE Clients (LECs) and BUS*

This list is provided to help you clarify the ATM connections that are established between the LE client and the BUS:

**Multicast Send VCC (bidirectional 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

An 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 required 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 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 sections describe each of these extensions.

## Broadcast Manager

Broadcast Manager (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 on to 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 advertisements and other broadcast requests. IPX routers and servers periodically broadcast their known network and service information. IPX clients send broadcast requests to locate a particular service or router. Generally, these broadcasts, called Routing Information Protocol (RIP) and Service Advertising Protocol (SAP) 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 and other broadcast requests to other IPX routers and servers. A broadcast frame managed by BCM IPX is sent as a series of unicast frames to the dynamically-learned set of IPX routers and servers.

When BCM IPX Server Farm Detection is enabled, BCM IPX will detect an IPX server farm when the number of IPX routers and servers discovered behind a given LEC exceeds a configurable threshold, the *BCM IPX Server Farm Threshold*. When a server farm is detected, BCM IPX broadcasts a managed frame to each LEC representing a server farm, rather than transmitting multiple unicast frames to each downstream IPX router and server in the server farm. BCM IPX can now intelligently use the broadcast mechanism in areas of the network where it is desirable to do so.

With BCM IPX 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 BCM static target. An example of such a device is a station running software that discovers the IPX network topology by monitoring IPX advertisements.

If BCM IPX Server Farm detection is enabled and you wish to prevent a particular LEC from being treated by BCM IPX as a Server Farm, configure a BCM static target with the LEC's ATM address and a MAC address of 00.00.00.00.00.00. This forces BCM IPX to send frames managed by BCM as multiple unicast frames to each downstream IPX router and server detected behind this LEC, even if the number of routers and servers detected exceeds the *BCM IPX Server Farm Threshold*.

## 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 provides support for NetBIOS Namesharing. That is, BCM NetBIOS handles OS/2 LANServer stations with multiple LAN adapters sharing the same NetBIOS name.

## 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 IP or BCM NetBIOS 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 an 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 19 provides a framework for describing the MSS redundancy solution. See the chapter entitled “Configuring”
Each LES/BUS may be independently configured for redundancy (the default is no

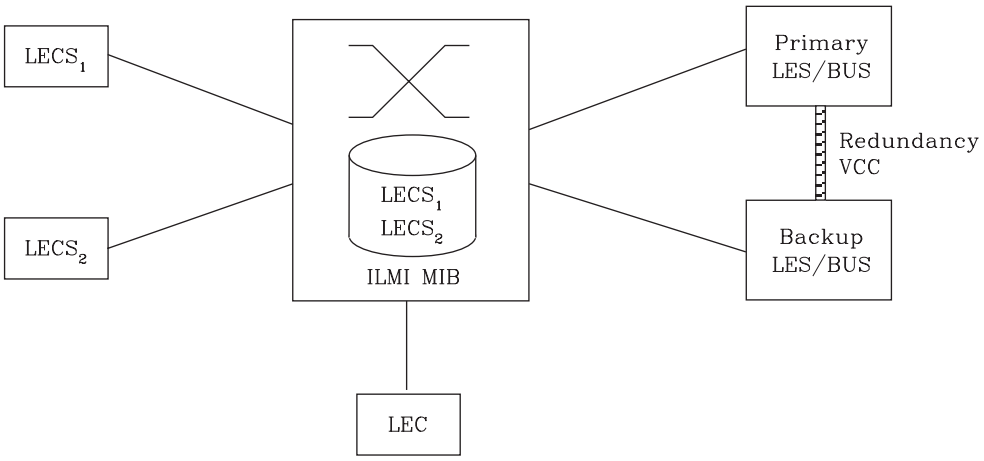



*Figure 19. LAN Emulation Redundancy*

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 returns either the primary or backup LES address.

The LECS returns the backup LES address 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 last assigned to the primary LES (within the past 5 minutes). Otherwise, it returns the primary LES address to the LE client.

The LECS retains a short-term memory of all client assignments so that it can alternately direct an 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. could be on MSS Server 1, while

## 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.

With the LECS interface, the router may ensure that an LE Client joins an ELAN only if the LECS approves of the join. This shifts the security burden from the LES to the LECS. Unfortunately, the LECS is also non-secure. The LECS accepts connections and queries from any station without verification. An intruder station may connect to the LECS and repeatedly query it for various configurations. The intruder may also pose as some other station and download another station's configuration.

LECS Access Controls permit the user to configure a list of ATM address prefixes which are not allowed access to the LECS configuration database. All LECS connection attempts and LE_CONFIGURE_REQUESTs from matching ATM addresses are automatically rejected. When used in conjunction with the LECS interface, a secure LANE environment is provided.

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 256 for more information.
2. Activate the LECS Interface on the router.
3. Activate the security option of the LESs.
4. Activate LECS Access Controls for any ATM address prefixes that should not be allowed to access the LECS.
5. 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 who may 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 an LE client, you only need to specify the ELAN type. If you define two LE clients on a single ATM interface and bridge them 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 you select 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, you must select the LECS ESI and configure a default ELAN assignment policy. See “Overview of the LECS Function” on page 256 for more information.

# Chapter 28. Using ATM

This chapter describes how to use the ATM interface. It includes the following sections:



## ATM and LAN Emulation

LAN emulation provides support for virtual Token-Ring and Ethernet LANs over an ATM network. Refer to "How to Enter Addresses" for a discussion of ATM addressing.

## How to Enter Addresses

Enter addresses 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

   Enter IP addresses in dotted decimal format, a 4-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

   Enter ATM addresses, MAC addresses, and route descriptors 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

   ```
   A1FF01020304
       or
   A1-FF-01-02-03-04
       or
   A1.FF.01.02.03.04
       or
   39.84.0F.00.00.00.00.00.00.00.00.00.03.10.00.5A.00.DE.AD.08
       or
   A1:FF:01:02:03:04
       or even
   A1-FF.01:0203:04
   ```

   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.

## ATM-LLC Multiplexing

Protocols that run natively over an ATM interface can use ATM-LLC multiplexing to share ATM addresses and both SVC and PVC channels between users. ATM-LLC is implicitly configured when the protocols are configured and can be monitored using the `ATM Config+` command prompt from **t 5**. There are no explicit configuration options for the ATM-LLC multiplexing function. For example, if two protocols which use ATM-LLC multiplexing are configured to use the same local ATM address (local endpoint), this implicitly configures ATM-LLC to use the same shared ATM address for both protocols.

See “ATM-LLC Monitoring Commands” on page 288 for additional information.

Sharing of ATM addresses or SVC/PVC channels is not possible between protocols that use the ATM-LLC multiplexing function and those that do not use the ATM-LLC multiplexing function (such as Classical IP). Currently, Server Cache Synchronization Protocol (SCSP) and APPN are the only two protocols that use the ATM-LLC multiplexing function.

## ATM Virtual Interface Concepts

An ATM Virtual Interface (AVI) creates the appearance of multiple ATM interfaces when, in fact, there is only one physical ATM interface. One or more AVIs can be configured for each physical ATM interface on the router. AVIs have the following characteristics:

- Each AVI must be defined on one (and only one) physical ATM interface. ATM real interface (ARI) will be used to mean a physical ATM interface.
- One or more AVIs can be configured on each ARI on a router.
- Higher layer protocols treat ARIs and AVIs equally. The protocols see the total number of ATM interfaces as the sum of the number of ARIs and AVIs configured on the router.
- Protocols can be configured on each ATM interface (real or virtual) independently of other interfaces.

  For example, one can configure IP on interface 0 (which is a real ATM interface) with IP address 9.1.1.1 and another instance of IP with address 9.2.1.1 on interface 1 (which is an AVI). Whether an interface is a real ATM interface or a virtual interface configured on a real interface makes no difference to the protocol (IP in the example). In addition, whether virtual interface 1 is configured on top of real ATM interface 0 or some other physical ATM interface is also transparent to the protocols.

## Advantages of Using ATM Virtual Interfaces

Major advantages of using the ATM Virtual Interfaces are:

- Using the ATM Virtual Interface feature increases the number of protocol instances that can be supported on a physical ATM interface.

  The actual number of AVIs that can be configured on an ARI is limited by physical resources, such as memory, available on the router. The total number of

interfaces that can be created depends on the data packet size for the interfaces and is limited to a maximum number of 253 per router.

The use of AVIs significantly improves the configuration options for protocols such as IPX that are limited to one instance or address per ATM interface. By configuring an appropriate number of AVIs, several IPX addresses can be supported on each physical ATM interface.

- The ATM Virtual Interface feature is crucial for supporting multicast routing protocols (such as MOSPF) over ATM networks.

  In order for multicast to operate correctly, each logical subnet ***must*** be configured on a different interface because multicast routing protocols typically function in such a way that a packet coming in from a router interface will never be sent out over the same interface. Thus, if more than one subnet is configured on an interface and a source in one subnet sends a multicast packet to a member in another subnet defined on the same interface, this member will never receive the packet.

  By creating an individual virtual interface for each subnet, packet multicasting can be performed successfully. Typically, the number of ATM interfaces on a router will be limited, in turn limiting the number of subnets that can be correctly configured for multicast operation. However, by creating as many AVIs as needed (according to the number of subnets that are required to be configured on the router), the number of physical ATM interfaces will no longer limit the number of subnets that can be configured on a router for correct multicast operation.

  For example, the “one-armed” router cannot support multicast traffic over interfaces other than ELANs without the AVI feature, because incoming packets will never be sent out the same interface and will be discarded instead.

- Creating multiple AVIs on an ARI and configuring each different protocol instance (for example, each IP subnet) on a different AVI on the same ARI, can improve performance.

  For example, when multiple subnets are configured on a single physical ATM interface, the interface will have to reduce the maximum transmission unit or MTU (the maximum packet size that can be sent or received over that interface) to the smallest MTU of all subnets sharing the same interface. However, if multiple AVIs are created on that ARI and each IP subnet is configured on a different AVI, every subnet can continue to use its existing MTU size without consideration of other subnets configured on the same physical ATM interface. This avoids possible reduction in throughput and delays due to packet fragmentation and reassembly caused by MTU size reduction.

  Another performance improvement can be achieved by distributing the number of protocol addresses configured on a physical interface over different virtual interfaces configured on the same physical interface. The per-interface protocol lists get shortened, resulting in faster searches and reduced processing time.

## Disadvantages of using ATM Virtual Interfaces

The disadvantages of using ATM Virtual Interfaces are:

- Because AVIs do not have any physical resources of their own, each virtual interface may have fewer Virtual Connections (VCs) than a single physical interface. The available resources (in this example VCs) are partitioned among the different virtual interfaces configured on a single ARI and the ARI itself.

  In the current implementation, resource allocation is *on demand*. Each physical ATM interface has a pool of resources that are available for use by all AVIs and the single ARI itself.

**Note:** Because all resources are shared among the ARI and all its AVIs, an ESI added on an ARI is automatically available to all AVIs configured on the ARI. You should not assign the same ESI and selector combination to two different protocol clients using the same ARI even though they are configured on different AVIs.

Limited PVC sharing is allowed across the ARI and the AVIs configured on the ARI. PVC sharing is limited to different protocol instances. Multiple instances of the same protocol are not allowed to share the same PVC.

# Chapter 29. Configuring and Monitoring ATM

This chapter describe the ATM interface configuration and operational commands. It includes the following sections:



## 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 4. The OPCON Process and Commands" on page 29.) For example:

   ```
   * talk 6
     Config>
   ```

   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 an 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:

   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 34. ATM Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |

### ATM Configuration Commands (Talk 6)

*Table 34. ATM Configuration Command Summary (continued)*

| Command | Function |
|---|---|
| INTERFACE | Displays the `ATM Interface Config>` prompt from which you can list, change, or configure the ATM Interface.<br>• Add an ESI.<br>• List the current configuration or list ESIs.<br>• Remove an ESI.<br>• Set parameters of the ATM network.<br>• Enable or disable an ESI.<br>• Exit |
| LE-CLIENT | Displays the `LE Client Config>` prompt from which you can list, change, or configure the LAN Emulation Client Interface as described in “Chapter 30. Using LAN Emulation Clients” on page 289 .<br>• Add a LAN Emulation Client (LEC) for a token-ring or Ethernet emulated LAN.<br>• Configure a LEC by network #. This command displays the `LE Config>` prompt, from which you can configure a specific LAN Emulation Client (LEC).<br>• List LAN Emulation Clients (LECs).<br>• Remove a LAN Emulation Client (LEC). |
| VIRTUAL ATM | Displays the `ATM Virtual Interface Config>` prompt from which you can list, add, or remove the ATM Virtual Interface as described in “ATM Virtual Interface Configuration Commands” on page 283 |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## ATM Interface Configuration Commands

This section summarizes and then explains the commands for configuring a specific ATM interface.

Enter the commands at the `ATM INTERFACE>` prompt.

*Table 35. ATM INTERFACE Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add | Adds an ESI. |
| List | Lists the current configuration or list ESIs. |
| Qos | Displays the `ATM I/F 0 QOS Config>` prompt from which you can configure Quality of Service as described in “QoS Configuration” on page 278. |
| Remove | Removes an ESI. |
| Set | Sets parameters of the ATM network. |
| Disable | Disables an ESI. |
| Enable | Enables an ESI. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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 to be unique.

**Syntax:**

**add** esi *esi-address*

**esi** *esi-address*

Address of End System Identifier.

**Valid Values:** Any 12 hexadecimal digits

**Default Value:**

none

## List

Use the **list** command to list the configuration of this ATM device or to list the set of configured ESIs.

**Syntax:**

**list** configuration

esi

**configuration**

Lists the ATM device configuration. For an explanation of the listed fields, see “Set” on page 278.

**Example: `list con`**

```
                        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
```

**esi** Lists the ESIs in the ATM configuration.

**Example: `list esi`**

```
ATM INTERFACE> list esi

        ESI                Enabled
------------------------   -------
000000000009                  YES
000000000100                  YES
```

## QoS Configuration

Use the **qos-configuration** command to display the `ATM I/F 0 QOS Config>` prompt from which you can configure Quality of Service as described in "QoS Configuration".

**Syntax:**

**qos-configuration**

## 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 that attempts to use a removed ESI may 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
network-id

**max-data-rate** *speed*
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. Signaled 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

100

155

**Default Value:**

155

**Example:**

```
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 signaling and switched connections. Increasing this parameter will allow more simultaneous SVCs. However, more system memory will be required to manage these connections.

**Valid Values:**

An integer in the range 64 - 10500

**Default Value:**

1024

**Example:**

```
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 64 – 1024

**Default Value:**

209

**Example:**

```
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 hierarchical 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 can 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 require you to explicitly configure a selector from the configured selector range. Other components, such as Classical IP clients, allow the selector to be automatically assigned at run-time. You do not have to choose the selector because the router does this when it activates. This selector is not guaranteed to be consistent across router restarts. Automatic selector assignment is useful only for those ATM components whose ATM address does not have to be already known 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:**

```
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. This includes all LECs and 1483 bridge interfaces.

**Valid Values:**
An integer in the range 16 – 32000

**Default Value:**
9234

**Example:**

```
ATM INTERFACE> set max-frame 1000
```

**max-mp-parties**

Sets the maximum number of leaves on a point-to-multipoint connection initiated 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 1 – 5000

**Default Value:**
512

**Example:**

```
ATM INTERFACE> set max-mp-parties 300
```

**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:**

```
ATM INTERFACE> set trace 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. If the UNI versions are configured on the ATM switch and ATM device interface to a specific version (not AUTO-DETECT), the UNI versions must match.

If the UNI version is configured as AUTO, the ATM device attempts to learn the UNI version to use from the switch.

In UNI AUTO-DETECT mode, if the switch does not respond to the query for UNI version, the default is UNI 3.0. If the switch responds with a value other than UNI 3.0 or UNI 3.1, the default is UNI 3.1.

**Valid Values:**
[UNI 3.0|UNI 3.1|AUTO-DETECT|None]

**Default Value:**
UNI 3.0

**Note:** Must be compatible with the ATM switch.

**Example:**

```
ATM INTERFACE> set uni-version 3.0
```

**network-id**

Sets the network id of the ATM interface. Multiple ATM interfaces should have the same network id if there is ATM connectivity between the interfaces.

**Valid Values:**

0 - 255

**Default Value:**

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 System Identifiers.

**Valid Values:**

Any 12 hexadecimal digits

**Default Value:**

none

**Example: `disable esi`**

```
ATM INTERFACE> disable esi 00:00:00:00:00:09
```

# Accessing the Virtual ATM Interface Configuration Process

From the `ATM Config>` prompt of a selected real ATM interface, use the **Virtual ATM** command to enter the Virtual ATM configuration command mode.

# ATM Virtual Interface Configuration Commands

This section summarizes the ATM virtual interface configuration commands. Enter the commands at the `ATM virtual interface config>` prompt.

*Table 36. ATM Virtual Interface Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add | Adds a virtual ATM interface. |
| List | Lists the current configured virtual ATM interfaces. |
| Remove | Removes the virtual ATM interface from the current configuration. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Add

Use the **add** command to add an ATM virtual interface. A new ATM virtual interface is added to the corresponding ATM real interface (the configuration menu from which this ATM virtual interface configuration menu is accessed). The net/interface number assigned to the newly created ATM virtual interface is displayed and it is one number greater than the current largest interface number.

**Syntax:**

**add**

**Example:**

```
ATM Virtual Interface config> add
Added ATM Virtual Interface Net as interface 5 on physical ATM interface 0
ATM Virtual Interface config>
```

## List

Use the **list** command to list configured ATM virtual interfaces defined on the current real ATM interface.

**Syntax:**

**list**

**Example:**

```
ATM Virtual Interface config> list

                  ATM Virtual Interface Nets
------------------------------------------------------------------------
  ATM interface number = 0
  ATM Virtual Interface Net interface number = 5

ATM Virtual Interface config>
```

## Remove

Use the **remove** command to delete an ATM virtual interface. The virtual ATM interface on the real ATM interface with the specified interface number will be

removed from the SRAM configuration records. If you do not specify an interface number, the last ATM virtual interface on this real ATM interface will be deleted. If you enter a question mark (?), all ATM virtual interfaces on the current real ATM interface will be listed and you can select from that list the interface you want to remove.

**Syntax:**

**remove** *n*

**Example:** `remove 5`

```
Virtual ATM 5 deleted successfully.
ATM Virtual Interface config>
```

## Accessing the ATM Monitoring Process

Use the following procedure to access the ATM monitoring 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 4. The OPCON Process and Commands" on page 29.) For example:

   ```
   * talk 5
   +
   ```

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

2. Enter **interface** at the + prompt to display a list of configured interfaces.
3. Record the interface numbers.
4. Enter **network** followed by the number of the ATM interface.

   ```
   + network 5
   ATM+
   ```

   The ATM monitoring prompt (`ATM+`) is displayed.

## ATM Monitoring Commands

This section summarizes the ATM monitoring commands for monitoring ATM interfaces. Enter the commands at the `ATM+` prompt.

*Table 37. ATM monitoring command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Interface | Displays the `ATM Interface+` prompt from which you can monitor the ATM Interface, as described in "ATM Interface Monitoring Commands (ATM INTERFACE+ Prompt)" on page 285 . |
| Atm-llc | Displays the `ATM LLC+` prompt from which you can monitor endpoints, a set of user clients, and a set of ATM channels. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Interface

Displays the `ATM Interface+` prompt, described in "ATM Interface Monitoring Commands (ATM INTERFACE+ Prompt)".

**Syntax:**

**interface**

## ATM-LLC

Displays the `ATM-LLC+` prompt, described in "ATM-LLC Monitoring Commands" on page 288 .

**Syntax:**

**atm-llc**

# ATM Interface Monitoring 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 38. ATM INTERFACE monitoring command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| List | Lists ATM addresses and VCCs. |
| 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 you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## List

Use the **list** command to list various categories of ATM data.

**Syntax:**

**list** addresses
all
circuit
vccs
reserved-bandwidth

**addresses**
Lists the ATM addresses, along with a descriptive name, in use on the device.

**Example:**

```
ATM INTERFACE+ list addresses

                ATM Address                          Name
----------------------------------------  ----------------------------------
39999999999999000099990200000041347391804  LEC 1 'eth1'
39999999999999000099990200000041347391802  LES/BUS 'eth1'
```

**all** Lists all of the following:

- Addresses
- Circuit statistics
- VCCs
- Reserved Bandwidth

**circuit** Lists the statistics for a particular VCC by specifying the particular VCI-VPI pair. You can also specify the circuit on the command line; for example: list circuit 33.

**Example:**

```
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:**

**P-P** point to point VCC

**P-MP** point to multipoint VCC

**ILMI** Interim Local Management Interface VCC

**SAAL** signalling VCC

**Bx-y** Internally bound VCC to VPI x, VCI y

**Sx-y** Internally spliced VCC to VPI x, VCI y

**reserved-bandwidth**
Lists the reserved bandwidth on the ATM Interface.

**Example:**

```
ATM INTERFACE+ list reserved-bandwidth
Line Rate                  : 155000 Kbps
Peak Reserved Bandwidth    : None
Sustained Reserved Bandwidth : None
```

# 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 175.

**Syntax:**

**trace** list
on

off

**list** Displays the current packet tracing options on the ATM interface.

**Example:**

```
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:**

```
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:**

```
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 wrap statistics are displayed.

For stop, the final wrap statistics are displayed.

**Syntax:**

**wrap** display
start
stop

**display**
Displays the current wrap settings.

**start** Starts the wrap procedure and specifies the VPI-VCI length of pattern and the pattern itself.

**Example:**

### ATM Interface Monitoring Commands (Talk 5)

```
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.

## ATM-LLC Monitoring Commands

This section explains the commands for monitoring ATM LLC multiplexing.

Enter the commands at the ATM-LLC+ prompt.

*Table 39. ATM LLC Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists various options |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to list various categories of ATM LLC monitoring data.

**Syntax:**

**list** endpoints

channels

**endpoints**

Lists the ATM addresses in use by protocols using the ATM-LLC multiplexing function on the device. The endpoint is displayed as the End System Identifier and the Selector.

**Example: `list endpoints`**

```
ATM-LLC+ list endpoints
```

**channels**

Lists the channels in use by protocols using the ATM-LLC multiplexing function on the device.

**Example: `list channels`**

```
ATM-LLC+ list channels
```

## ATM Virtual Interface Monitoring Commands

Monitoring the ATM virtual interface is done using the ATM LLC monitoring commands. See “ATM-LLC Monitoring Commands” for additional information.

# Chapter 30. Using LAN Emulation Clients

This chapter describes LAN Emulation Clients (LECs). It includes the following sections:

- "LAN Emulation Client Overview"

## 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 ports 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" on page 291
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.

An explanation of commands for LAN Emulation Clients follows.

# Chapter 31. Configuring and Monitoring LAN Emulation Clients

This chapter describes how to configure LAN Emulation Clients (LECs). It includes the following sections:



## Configuring LAN Emulation Clients

This section summarizes and explains the commands for configuring and using the set of LE Clients on a particular ATM interface.

To get to the `LE Client Config>` prompt, enter **le-c** at the ATM Config> prompt as described in "ATM Configuration Commands" on page 275.

Enter the commands at the `LE Client Config>` prompt under the `ATM Config>` prompt, as described in "ATM Configuration Commands" on page 275.

*Table 40. LAN EMULATION Client Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Add | Adds a LEC for the following types of ATM Forum-compliant Emulated LANs architectures:<br>• Ethernet<br>• Token Ring |
| Config | Gets you to the `LEC Config>` prompt, from which you can configure a specific LAN Emulation Client. |
| List | Lists the LEC. |
| Remove | Removes a LEC. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## 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 Emulated LAN as interface 3
```

### LE Client Config>

**ethernet**
Ethernet emulated LAN

**Example: `add ethernet`**

```
LE Client Config> add ethernet
Added Emulated LAN 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.

**Syntax:**

**config** interface#

**interface#**
An integer number assigned by the router when the LEC was added to the configuration. Use the **list** command to determine the interface number assigned to the LEC.

**Example: `config`**

```
LE Client Config> config 3
ATM LAN Emulation Client configuration
```

## List

Use the **list** command to list the LAN emulation clients.

**Syntax:**

**list**

**Example: `list`**

```
LE Client Config> list
                 ATM Forum Compliant Emulated LANs
-----------------------------------------------------
  Physical ATM interface number = 0
  LEC interface number = 1
  Emulated LAN type    = Token Ring Forum Compliant
  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:**

**remove** interface#

**interface#**
An integer number assigned by the router.

## Configuring an ATM Forum-Compliant LE Client

This section explains the commands for configuring an ATM Forum-compliant LAN Emulation Client. Enter the appropriate commands at either the `Ethernet Forum Compliant LEC Config>`prompt or the `Token Ring Forum Compliant LEC Config>`prompt. Commands in the following table apply to both Token-Ring and Ethernet LECs except where indicated.

Enter the commands at the `LEC Config>` prompt after entering the **config** command at the `LE Client Config>` prompt.

*Table 41. LAN Emulation Client Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| ARP-Configuration | Allows you to configure the LE-ARP configuration for the ATM Forum-compliant client |
| RIF-Timer | Sets the maximum amount of time that information in the RIF is maintained before it is refreshed. Applies only to Token-Ring LECs. |
| Source-routing | Used to enable or disable source-route bridging. Applies only to Token-Ring LECs. |
| IP-Encapsulation | Sets the IP encapsulation as Ethernet (type X’0800’) or IEEE (802.3 with SNAP). Applies only to Ethernet LECs. |
| List | Lists the LAN Emulation Client configuration. |
| QOS-Configuration | Gets you to the `elan-x LEC QoS Config>` prompt from which you can configure Quality of Service as described in “LE Client QoS Configuration Commands” on page 855. |
| Set | Sets the LAN Emulation Client parameters. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## ARP Configuration

Use the **arp-configuration** command to configure the static LE-ARP entries for the ATM forum-compliant LAN Emulation Client.

**Syntax:**

**arp-configuration**

**Example:**

```
Token Ring Forum Compliant LEC Config> arp-configuration
ATM LAN Emulation Clients ARP configuration
```

*Table 42. ATM LAN Emulation Client ARP Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add | Adds an LE-ARP cache entry using a MAC or route descriptor ARP. |
| Config | Sets cache entry QOS parameter values. |

*Table 42. ATM LAN Emulation Client ARP Configuration Commands Summary (continued)*

| Command | Function |
|---|---|
| List | Lists configured ARP cache entries. |
| Remove | Removes an ARP cache entry. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Add

Use the **add** command to add an ARP cache entry using the MAC address or a route descriptor.

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 (:).

**Syntax:**

**add** mac
route-descriptor

**Example 1:**

```
ARP config for LEC>add mac
MAC address of LE ARP Entry []? 123456789098
ATM address in 00.00.00.00.00.00:... form []? 390f000000000000000000000000123456789098
Destination Type -  REMOTE or LOCAL [Remote]?
```

**Example 2:**

```
ARP config for LEC>add route 12.34
ATM address in 00.00.00.00.00.00:... form []? 390f000000000000000000001234567890988888
ARP config for LEC>
```

## Config

Use the **Config** command to configure the permanent ARP cache entry QOS parameters for the ATM forum-specific LAN Emulation Client.

**Syntax:**

**config** *arp-entry-number*

**Example:**

```
ARP config for LEC> config
ARP entry number [1]
Configure LEC ARP entry
```

*Table 43. ATM LAN Emulation Client ARP Config Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Set | Sets QOS parameter values. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

***Set:***

Use the **Set** command to configure the permanent ARP cache entry QOS parameters for the ATM forum-specific LAN Emulation Client.

**Syntax:**

| | |
|---|---|
| **set** | max-reserved-bandwidth |
| | traffic-type |
| | peak-cell-rate |
| | sustained-cell-rate |
| | qos-class |
| | max-burst-size |

**Example:**

```
ARP entry 'identifier' config> set ?
 MAX-RESERVED-BANDWIDTH
 TRAFFIC-TYPE
 PEAK-CELL-RATE
 SUSTAINED-CELL-RATE
 QOS-CLASS
 MAX-BURST-SIZE
```

See “Chapter 71. Using Quality of Service (QoS)” on page 847 for detailed information about the QoS parameters.

### List

Use the **list** command to display information about ARP configuration.

### Remove

Use the **remove** command to remove an configured MAC address or Route Descriptor LE-ARP entry.

Select the ARP entry number to be removed from the list provided.

**Syntax:**

| | |
|---|---|
| **remove** | *arp-entry-number* |

## RIF-Timer (for Token-Ring Forum-compliant LEC only)

Use the **RIF-Timer** command to set the maximum amount of time that information in the RIF is maintained before it is refreshed. Range is 0 to 4096. The default is 120 seconds.

**Syntax:**

| | |
|---|---|
| **rif-timer** | *value* |

**Example:**

```
rif-timer 100
```

## Source-routing (for Token-Ring Forum-compliant LEC only)

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 route bridge.

This function of the device is not changed whether source routing is enabled or disabled. The default setting is enabled.

Some stations cannot properly receive frames with 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 through which you want to bridge IP, IPX, and AppleTalk Phase 2 packets. Source routing must also be enabled so that LLC test response messages can be returned.

**Syntax:**

| | |
|---|---|
| **source-routing** | enable |
| | disable |

**Example:**

```
source-routing disable
```

## IP-Encapsulation (for Ethernet ATM Forum-compliant LEC only)

Use the **IP-encapsulation** command to select Ethernet (Ethernet type X'0800') or IEEE 802.3 (Ethernet 802.3 with SNAP). Specify either type **E**thernet or **I**EEE-802.3.

**Syntax:**

| | |
|---|---|
| **IP-encapsulation** | Ethernet |
| | IEEE-802.3 |

## List

Use the **list** command to list the LE client configuration.

**Syntax:**

**list**

## QoS

Use the **qos-configuration** command to get you to the `LEC QoS Config>` prompt from which you can configure Quality of Service as described in "LE Client QoS Configuration Commands" on page 855.

**Syntax:**

**qos-configuration**

## Set

Use the **set** command to set LE Client parameters.

**Syntax:**

**set**
- arp-aging-time
- arp-cache-size
- arp-queue-depth
- arp-response-time
- auto-config
- best-effort-peakrate
- bus-connect-retries
- conn-completion-time
- control-timeout
- elan-name
- esi-address
- flush-timeout
- forward-delay
- forward-disconnect-timeout
- frame-size
- initial-control-timeout
- lecs-atm-address
- les-atm-address
- mac-address
- multicast-send-avg
- multicast-send-peak
- multicast-send-type
- multiplier-control-timeout
- path-switch-delay
- reconfig-delay-min
- reconfig-delay-max
- retry-count
- selector
- trace
- unknown-count
- unknown-time
- vcc-timeout

**arp-aging-time**
Sets ARP 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 network configuration.

**Valid Values:**
An integer number of seconds in the range of 10 to 300.

**Default Value:**
300

**Example:**

```
LEC Config> set arp-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 permit the client to simultaneously converse with a larger number of destinations.

**Valid Values:**
An integer number in the range of 10 to 65535.

**Default Value:**
5000

**Example:**

```
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 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:**

```
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 increase 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:**

```
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.

**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 301.

**Default Value:**

NO

**Example:**

```
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 1 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.

**Valid Values:**

An integer number in the range of 1 - device maximum data rate.

**Default Value:**

155000

**Example:**

```
LEC Config> set best-effort-peakrate 24000
```

**bus-connect-retries**

This parameter sets the maximum number of times that the LEC will attempt to reconnect to the BUS before returning to the initial state.

**Valid Values:**

0 - 2

**Default Value:**

1

**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.

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

**Example:**

```
LEC Config> set connection-completion-time 5
```

**control-timeout**
This parameter sets the maximum cumulative control timeout of a request.

A current timeout value is initialized to the value of ***initial-control-timeout***. If a response to a request is not received within the current timeout value, the current timeout is multiplied by the value of the ***multiplier-control-timeout*** and the request is reissued. Each time the current timeout value expires, this process is repeated until the current timeout value exceeds the value of ***control-timeout***.

**Valid Values:**
An integer number of seconds in the range of 10 to 300.

**Default Value:**
30

**Example:**

```
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:**

```
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 device.

**Valid Values:**
Any 12 hexadecimal digits.

**Default Value:**
Burned-in ESI

**Example:**

```
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. During recovery, any queued frames are dropped and a new flush request is sent.

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.

**Valid Values:**

An integer number of seconds in the range of 1 to 4.

**Default Value:**

4

**Example:**

```
LEC Config> set flush-timeout 3
```

**forward-delay**

Sets the forward delay. Entries in the LE ARP cache must be periodically re-verified. 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 re-verification traffic.

**Valid Values:**

An integer number of seconds in the range of 4 to 30.

**Default Value:**

15

**Example:**

```
LEC Config> set forward-delay 10
```

**forward-disconnect-timeout**

This parameter sets the amount of time that a LEC will wait after losing its last Multicast Forward VCC from the BUS before returning to the initial state. This delay permits the BUS to attempt to reconnect to the client without returning to the initial state.

**Valid Values:**

10 - 300 seconds

**Default Value:**

60

**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 280.

**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:**

```
LEC Config> set frame-size 4544
```

**initial-control-timeout**

This parameter sets the value of the initial control timeout used in the control timeout algorithm described in 300.

**Valid Values:**

1 - 10

**Default Value:**

5

**Example:**

```
LEC Config> set initial-control-timeout 10
```

**lecs-atm-address**

Specifies the ATM address of the 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.

No default is provided.

**Note:** This command should be entered on one command line. It is shown here on two lines because of spacing.

**Example:**

```
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 auto-config**command on page 298.

- If auto-config is YES, the les-atm-address is not configurable.
- If auto-config is NO, then the les-atm-address is required.

Specify the ATM address of the LES. No default is provided.

**Note:** This command should be entered on one command line. It is shown here on two lines because of spacing.

**Example:**

```
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 of the ATM interface, or you may specify a different MAC address. If you have two clients that are bridged together, they should use different MAC addresses.

This MAC address is registered with the LES when the client joins the ELAN.

**Valid Values:**

Any valid MAC address.

**Default Value:**

none

**Example:**

```
LEC Config> set mac-address
  Use adapter address for MAC? [No]
  MAC address []: 10.00.5a.00.00.01
```

**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 reserved bandwidth. 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 be less than or equal to 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 reserved, then multicast-send-avg and multicast-send-peak must be specified.

**Example:**

```
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 bandwidth. 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 be less than or equal to 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 reserved, then multicast-send-avg and multicast-send-peak must be specified.

**Example:**

```
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 reserved, then multicast-send-no and multicast-send-peak must be specified.

**Valid Values:**

Best Effort or Reserved

**Default Value:**

Best Effort

**Example:**

```
LEC Config> set multicast-send-type best-effort
```

**multiplier-control-timeout**

This parameter sets the value of the control timeout multiplier used in the control timeout algorithm described in 300.

**Valid Values:**

2 - 5

**Default Value:**

2

**Example:**

```
LEC Config> set multiplier-control-timeout 5
```

**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

**Example:**

```
LEC Config> set path-switch-delay 5
```

**reconfig-delay-min**

This parameter sets the minimum delay time when LEC returns to the initial state. This value must be ≤ ***reconfig-delay-max***.

**Valid Values:**

1 - the value of ***reconfig-delay-max***

**Default Value:**

1

**Example:**

```
LEC Config> set reconfig-delay-min  5
```

**reconfig-delay-max**

This parameter sets the maximum delay time when LEC returns to the initial state. This value must be ≥ ***reconfig-delay-min***.

**Valid Values:**

1 - 10

**Default Value:**

5

**Example:**

```
LEC Config> set reconfig-delay-max 9
```

**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:**

```
LEC Config> set retry-count 2
```

**selector**

Specifies the selector portion of the client's ATM address. The combination of ESI and selector must be unique among all LANE components on the device. By default, a unique selector is selected for the configured ESI.

**Valid Values:**

Any octet, in hexadecimal, that is not in use by another LANE component with the same ESI.

**Example:**

```
LEC Config> set selector 01
```

**trace** Enables 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

**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:**

```
LEC Config> set unknown-time 5
```

**vcc-timeout**
Sets the VCC timeout. Data direct VCCs over which no traffic has been sent for this period of time should be released.

**Valid Values:** 0 to 31536000 seconds (1 year).

**Default Value:** 1200

**Note:** This parameter is meaningful only for SVC connections.

**Example:**

```
LEC Config> set vcc-timeout 1000
```

## Accessing the LEC Monitoring Environment

Use the following procedure to access the LEC monitoring 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 4. The OPCON Process and Commands” on page 29.) 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:

```
+ 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 Monitoring Commands

This section summarizes and then explains the LEC monitoring commands. You can access LEC monitoring commands at the LEC+ prompt. Table 44 shows the commands.

*Table 44. LE Config monitoring command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists:<br>• LEC Address Resolution Table (ARP)<br>• LEC configuration<br>• Data Direct VCC information<br>• LEC statistics<br>• VCC table. |
| MIB | Displays LEC MIB objects including:<br>• LEC MIB Configuration Table<br>• LEC MAC ARP Table<br>• LEC Route Descriptor Table<br>• LEC MIB Server VCC Tables<br>• LEC MIB Statistics Table<br>• LEC MIB Status Table |
| QoS | Gets you to the LEC x QoS+ prompt from which you can monitor Quality of Service as described in “Quality of Service Monitoring Commands” on page 862. |
| Trace | Sets packet tracing on or off. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

# 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:**

**list** arp-table
configuration
data-direct-vccs
statistics
vcc-table

**arp** Lists the LEC Address Resolution Table (entries in the ARP cache).

**Example:**

```
LEC+ list arp

         LEC Address Resolution (LE ARP Cache) Table

  Max Table Size          = 10
  Free Table Entries      = 10
  Current Mac Entries     = 0
  Current RD Entries      = 0
  Arp Aging Time          = 300
  Verify Sweep Interval   = 60

                              Xmit  BUS   Arp
                         Conn   Queue Frame Retry Aging
 MAC Address     Remote Handle Depth Count Count Timer  Destination ATM Ad
                                                                    dress
 ------------------------------------------------------------------------
 40.00.00.00.00.09 False 652   0     0     0     60     39.99.99.99.99.99.
                               99.00.00.99.99.30.02.40.00.00.00.00.09.81
```

**Note:** The Sweep Interval is always one-fifth of the ARP Aging Timer value.

**Max Table Size**
The total number of entries available

**Free Table Entries**
The number of free entries

**Current MAC Entries**

**Current RD Entries**
Route Descriptor ATM entries

**ARP Aging Time**
Time for an entry to be aged out

**Verify Sweep Interval**

**MAC Address**

**Remote**

**Connection Handle**

**Queue Depth**

### Xmit Frame Count

### BUS Retry Count

### ARP Aging Timer

### Destination ATM Address

## configuration

Lists the LEC configuration.

For Ethernet:

**Example:**

```
IBM LEC+ list config
                ATM IBM LEC Configuration
   Physical ATM interface number            = 0
   LEC interface number             = 7
   Primary ATM address
          ESI address               = Use burned in addr
          Selector byte             = 0x3
   Emulated LAN type                = Ethernet IBM
   Maximum frame size               = 1523
   LE Client MAC address            = Use burned in addr
   LE Server ATM address            = 00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00
   Forward Peak Rate                = 155000
   Backward Peak Rate               = 155000
    MAC cache size                   = 32
   MAC cache aging period           = 60
   Route Descriptor cache size      = 32
   Route Descriptor aging period    = 60
   LES Registration interval        = 60
   LES Registration retry count     = 3
   LES keep alive count             = 10
   Packet trace                     = No
   IP Encapsulation                 = ETHER
```

For Token Ring:

**Example:**

```
IBM LEC+list config

                     ATM IBM LEC Configuration
   Physical ATM interface number             = 0
   LEC interface number             = 10
   Primary ATM address
          ESI address               = Use burned in addr
          Selector byte             = 0x6
   Emulated LAN type                = Token Ring IBM
   Maximum frame size               = 4551
   LE Client MAC address            = Use burned in addr
   LE Server ATM address            = 39.84.07.00.00.00.00.00.00.00.00.00.01.10.00.5A.DD.DA.02
   Forward Peak Rate                = 155000
   Backward Peak Rate               = 155000
    MAC cache size                   = 32
   MAC cache aging period           = 60
   Route Descriptor cache size      = 32
   Route Descriptor aging period    = 60
   LES Registration interval        = 60
   LES Registration retry count     = 3
   LES keep alive count             = 10
   Packet trace                     = No
   RIF Aging Timer                  = 120
   Source Routing                   = Enabled
```

**Example:**

```
LEC+ list config

   Physical ATM interface number = 0
   LEC interface number          = 9
   LEC ATM address               = 39.99.99.99.99.99.99.00.00.99.99.31.01.09.FC.DD.D0.32.70.0A
   LEC MAC address               = 40.00.82.10.17.09
```

```
lecConfigMode                 = Manual
lecConfigLanType              = 802.5 - Token Ring
lecConfigMaxDataFrameSize     = 4544
lecConfigLanName              =
lecConfigLesAtmAddress        = 39.99.99.99.99.99.99.00.00.99.99.31.01.40.00.82.10.17.00.09
lecControlTimeout             = 30
lecMaxUnknownFrameCount       = 10
lecMaxUnknownFrameTime        = 1
lecVccTimeoutPeriod           = 1200
lecMaxRetryCount              = 1
lecAgingTime                  = 300
lecForwardDelayTime           = 15
lecExpectedArpResponseTime    = 1
lecFlushTimeout               = 4
lecPathSwitchingDelay         = 6
lecLocalSegmentId             = 0x0
lecMulticastSendType          = 1
lecMulticastSendAvgRate       = 365566
lecMulticastSendPeakRate      = 365566
lecConnectionCompleteTimer    = 4
lecInitialControlTimeout      = 5
lecControlTimeoutMultiplier   = 2
V2 Capable                    = TRUE
lecForwardDisconnectTimeout   = 60
lecMinReconfigDelay           = 1
lecMaxReconfigDelay           = 5
lecMaxBusConnectRetries       = 0
lecElanId                     = 0
ExplorerExclude               = TRUE
LE ARP queue depth            = 5
LE ARP cache size             = 5000
Forward peakrate              = 365566
Backward peakrate             = 365566
Packet trace                  = Off
RIF aging timer               = 120
Source Routing                = enabled
```

See “Set” on page 297 for a definition of the parameters shown in the above examples.

**data** Lists the LEC Data Direct VCC information.

**Example:**

```
LEC+ list data

          LEC Data Direct VCC Table

   Max Table Size     = 1019    Max no of SVC connections
   Current Size       = 0       Currently used
   Inactivity Timeout = 1200    No Data Xfer Timeout before connection is
                                closed (seconds)
   Sweep Interval     = 60
      Conn              Inactive  User
      Handle  VPI  VCI   Timer    Count  Destination ATM Address
    ------------------------------------------------------------------------

       652     0  7241    300       1    39.99.99.99.99.99.99.00.00.99.99.30.02.
                                         40.00.00.00.00.09.81


    ------------------------------------------------------------------------
```

**statistics**

Lists LEC statistics.

**Example:**

```
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
```

```
In Frames          = 0
Out Frames         = 0
In Bytes           = 0
Out Bytes          = 0
```

**VCC table**
Lists VCC table.

**Example:**

```
LEC+ list vcc
```

## 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:**

**mib** config-table
mac-arp-table
rd-arp-table
server-vcc-table
statistics-table
status-table

**config** Displays the LEC MIB Configuration Table.

**Example:**

```
LEC+ mib config

 lecConfigTable:
 lecConfigMode                 = Manual
 lecConfigLanType              = 802.3 - Ethernet
 lecConfigMaxDataFrameSize     = 1516
 lecConfigLanName              =
 lecConfigLesAtmAddress        = 39.84.0F.00.00.00.00.00.11.23.24.24.24.24.55.66.77.88.99.00
 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 number 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:**

```
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:**

```
LEC+ mib statistics

lecStatisticsTable:
   lecArpRequestsOut             = 1
   lecArpRequestsIn              = 0
   lecArpRepliesOut              = 0
   lecArpRepliesIn               = 1
   lecControlFramesOut           = 2
   lecControlFramesIn            = 2
   lecSvcFailures                = 1
```

**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:**

```
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.

   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.

   lecProxyClient                = False                Is LES acting like a
                                                        proxy ?
```

# QoS Information

Use the **qos-information** command to get to the `LEC x QoS+` prompt from which you can monitor Quality of Service as described in “Quality of Service Monitoring Commands” on page 862.

**Syntax:**

**qos-information**

# Chapter 32. Using Channel Adapters

This chapter describes how to plan for host definition and 2216 ESCON and Parallel Channel Adapter (PCA) support. It includes the following sections:

- "Host Definition Planning", provides information to help you plan for host definition.
- "Planning for 2216 Support" on page 335, describes considerations for support of the 2216 and channel adapters in a network.
- "Channel Adapter Overview" on page 336, describes channel adapter support:
  - LCS (LAN Channel Station) over which you can run TCP/IP
  - LSA (Link Services Architecture) over which you can run hierarchical SNA, including DLSw, APPN ISR, or APPN HPR
  - MPC+ (Multi-Path Channel) over which you can run APPN HPR, TCP/IP, and HPDT UDP (UDP+)

    **Note:** UDP+ is not supported on a Parallel Channel Adapter (PCA).
- "Configuring the Channel Adapter Interface" on page 352
- "Channel Adapter Configuration Commands" on page 356

## Host Definition Planning

This section provides information to help you plan for host definition. It includes information for system definition from the host perspective and information for definition from the 2216 perspective.

Before you can attach the 2216 to a channel, the host system must be configured correctly. The following series of steps is required to define the 2216 connection to the host. These definition steps should be done by your system programmer.

1. Define the 2216 to the host channel subsystem using either the host Input/Output Configuration Program (IOCP) or Hardware Configuration Definition (HCD) program.
2. Define the 2216 as a control unit to the host operating system.
3. Define the 2216 and configuration to the host program (TCP/IP or VTAM).

After the host definitions are complete, you must configure the 2216 channel interfaces using the command line interface, or using the configuration program described in *Configuration Program User's Guide*, GC30-3830. Many of the parameters which you provide when you configure the 2216 must match corresponding parameters in the host definition.

Finally, the stations will need to be configured to communicate through the 2216 to the host applications.

The following sections describe host definition and provide sample host configuration statements.

# IOCP Definition for the 2216

The following sections describe examples of IOCP definitions for the 2216 with channel adapters. The output of the IOCP device definitions (I/O Configuration Data Set or IOCDS) can be generated using MVS, VM, VSE, or in a stand-alone environment. Refer to the *ES/9000 and ES/3090 Input/Output Configuration Program User's Guide Volume A04*, GC38-0097, for details.

## Example IOCP Definition for the ESCON Channel

Figure 20 shows an example of an ESCON configuration. The S/390 host is divided into two logical partitions (LP): LPA and LPB. A connection on path 30 is configured between LPA and 2216A through ESCD switch 00. LPA is attached to ESCD port C0 and 2216A is attached to port C1. The connection between port C0 and C1 is dynamic.

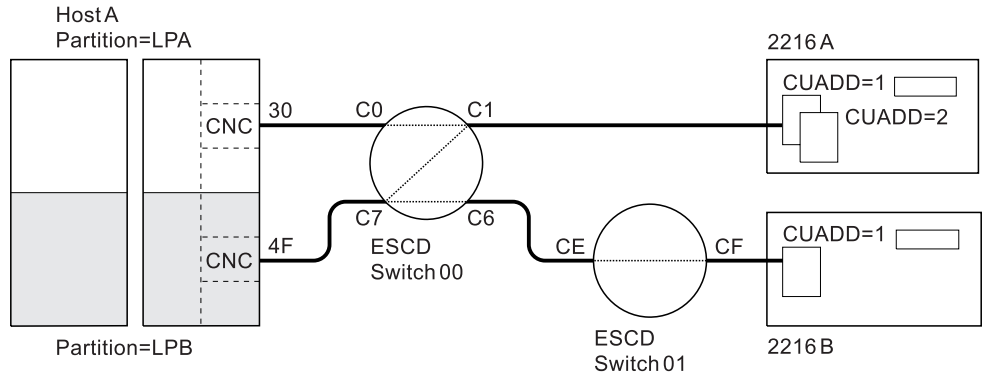



*Figure 20. ESCON Channel Configuration Example*

LPB on path 4F has a connection with 2216A through ESCD switch 00, and a connection with 2216B through ESCD switches 00 and 01. The connection between ports C7 and C6 is dynamic; the connection between ESCD ports CE and CF is dedicated.

The following example definitions match Figure 20:

Channel path definitions:

```
CHPID      PATH=((30)),TYPE=CNC,PART=(LPA),SWITCH=00
CHPID      PATH=((4F)),TYPE=CNC,PART=(LPB),SWITCH=00
```

Control unit and device definition for the 2216, with logical addressing = 1 for 2216A:

```
CNTLUNIT   CUNUMBR=500,PATH=30,UNIT=3172,LINK=C1,         X
           UNITADD=(00,32),CUADD=1
IODEVICE   ADDRESS=(500,32),CUNUMBR=500,UNIT=3172,        X
           UNITADD=00
```

Control unit and device definition for the 2216 with logical addressing = 2 for 2216A:

```
CNTLUNIT   CUNUMBR=600,PATH=4F,UNIT=3172,LINK=C1,         X
           UNITADD=(00,32),CUADD=2
IODEVICE   ADDRESS=(600,32),CUNUMBR=600,UNIT=3172,        X
           UNITADD=00
```

Control unit and device definition for the 2216, with logical addressing = 1 for 2216B:

```
CNTLUNIT   CUNUMBR=620,PATH=4F,UNIT=3172,LINK=C6,          X
           UNITADD=(20,32),CUADD=1
IODEVICE   ADDRESS=(620,32),CUNUMBR=620,UNIT=3172,         X
           UNITADD=20
```

The IOCP macroinstructions in the example:

- Assign a CHPID to logical partitions LPA and LPB.
- Define channel path 30 to the 2216 for partition LPA and channel path 4F for partition LPB.
- Identify channel type as an ESCON channel (CNC).
- Assign the two CHPIDs to ESCD switch number 00.
- Associate control unit numbers 500 and 600 to logical addresses 1 and 2 on 2216A and control unit number 620 to logical address 1 on 2216B.
- Assign link address C1 to control units 500 and 600 and link address C6 to control unit 620.
- Define unit addresses (subchannels) 00 through 1F to control units 500 and 600 and unit addresses 20 through 3F to control unit 620.
- Identify each control unit as a 3172 device.

**Considerations:**

1. The allowable device address range is 00 through FF. The 2216 address range is limited to 32 addresses, and only requires that the addresses defined at the host computer map to the address or addresses configured in the 2216. The address range can extend beyond the addresses actually used, but cannot overlap addresses of other control units cabled to the same CHPID or channel.
2. The ESCON channel mode of operation can be type CNC for basic ESCON channel mode or CVC if there is an ESCON Converter attached.
3. The IODEVICE UNIT parameter should be set to 3172.
4. The LINK number specifies the link address (ESCD port number) to which the 2216 is connected. When two ESCDs are connected in series, the link address must be the port number of the ESCD that has the dynamic connection and to which the 2216 is attached.
5. The logical address (CUADD) must be unique for a given path between a host channel and a 2216.

## Example IOCP Definition for the EMIF Host

Figure 21 on page 318 shows an example of an ESCON configuration using the ESCON Multiple Image Facility (EMIF). The S/390 host is divided into two logical partitions (LP): LPA and LPB. Both LPA and LPB are connected on path 30 to 2216 A through switch 00.

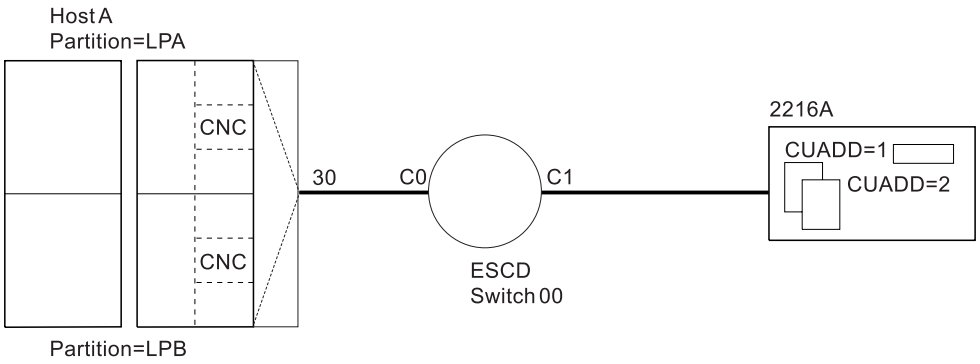



*Figure 21. EMIF Host Configuration Example*

The following example definitions match Figure 21:

Channel path definitions:

```
CHPID      PATH=((30)),TYPE=CNC,PART=(LPA,LPB),SWITCH=00
```

Control unit and device definition for the 2216, with logical addressing = 1 for 2216A:

```
CNTLUNIT   CUNUMBR=500,PATH=30,UNIT=3172,LINK=C1,         X
           UNITADD=(00,32),CUADD=1
IODEVICE   ADDRESS=(500,32),CUNUMBR=500,UNIT=3172,        X
           UNITADD=00
```

Control unit and device definition for the 2216, with logical addressing = 2 for 2216A:

```
CNTLUNIT   CUNUMBR=620,PATH=30,UNIT=3172,LINK=C1,         X
           UNITADD=(20,32),CUADD=2
IODEVICE   ADDRESS=(620,32),CUNUMBR=620,UNIT=3172,        X
           UNITADD=20
```

The IOCP macroinstructions in the example:

- Assign a CHPID to logical partitions LPA and LPB
- Define channel path 30 to the 2216 to be shared by partition LPA and partition LPB.
- Identify channel type as an ESCON channel (CNC)
- Assign the CHPID to ESCD switch number 00
- Associate control unit numbers 500 to logical address 1 and 620 to logical address 2 on 2216A
- Assign link address C1 to control units 500 and 620
- Define unit addresses (subchannels) 00 through 1F to control unit 500 and 20 through 3F to control unit 620
- Identify each control unit as a 3172 device.

**Considerations:**

1. The allowable device address range is 00 through FF. The 2216 address range is limited to 32 addresses, and only requires that the addresses defined at the host computer map to the address or addresses configured in the 2216. The address range can extend beyond the addresses actually used for the 2216, but cannot overlap addresses of other control units cabled to the same CHPID or channel.
2. The ESCON channel mode of operation can be type CNC for basic ESCON channel mode or CVC if there is an ESCON Converter attached.
3. The IODEVICE UNIT parameter should be set to 3172.

4. The LINK number specifies the link address (ESCD port number) to which the 2216 is connected. When two ESCDs are connected in series, the link address must be the port number of the ESCD that has the dynamic connection and to which the 2216 is attached.
5. The logical address (CUADD) must be unique for a given path between a host channel and a 2216.
6. Each partition must have a unique logical address defined on the 2216.

## Example IOCP Definition for the Parallel Channel Adapter (PCA)

Figure 22 shows an example of a simple Parallel Channel Adapter configuration where the channel type is identified as a block multiplexer channel, TYPE=BL, on path 5.

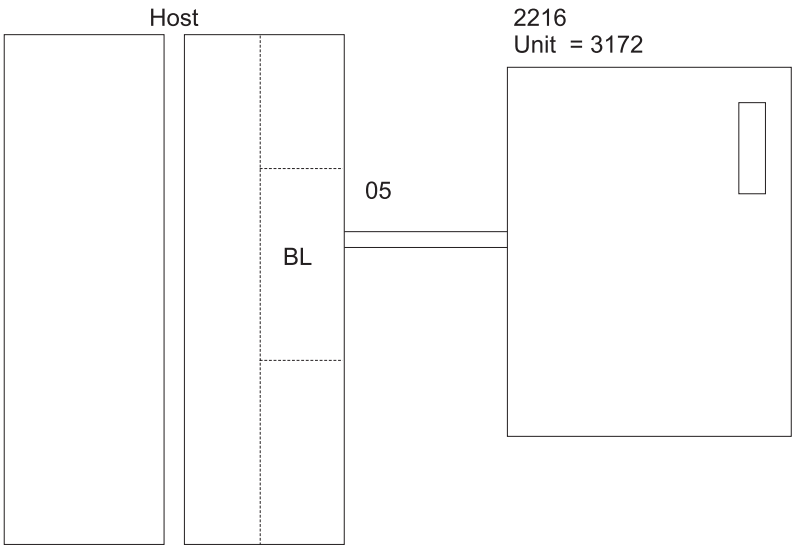



*Figure 22. Parallel Channel Adapter Configuration Example*

The IBM 2216 I/O device unit appears to the host as 3172 and must be defined as a 3172.

The following example definitions match Figure 22:

```
CHPID      PATH=((05)),TYPE=BL
CNTLUNIT   CUNUMBR=640,PATH=05,                          X
           PROTOCL=S4,UNIT=3172,                         X
           SHARED=N,UNITADD=((40,32))
IODEVICE   UNIT=3172,ADDRESS=((640,32)),                 X
           STADET=N,CUNUMBER=640,TIMEOUT=Y
```

The IOCP macroinstructions in the example:

- Identify channel type as a block multiplexer channel (BL).
- Name channel path 05 to which the PCA is attached.
- Assign the control unit number 640 to the PCA.
- Specify that the host channel supports up to 4.5 MB data-streaming (S4).
- Identify the control unit as type 2 where multiple I/O request are supported concurrently (SHARED=N).
- Define unit addresses 40 through 5F to the PCA (a range of 32 addresses).
- Identify the PCA control unit as a 3172-type device.

**Considerations:**

1. The address range for each PCA must be contiguous pairs of addresses for TCP/IP, a single address for VTAM and at least one read subchannel and one write subchannel for MPC+.

   The allowable device address range is 00 through FF. Each 2216 PCA can support a maximum of 32 subchannels. The 2216 PCA does not require a range of 32 addresses, it only requires that the addresses defined at the host computer map to the address or addresses configured for the 2216 PCA. The addresses used cannot overlap addresses used by other control units or PCAs cabled to the same CHPID or channel.
2. The PROTOCL parameter can be set to the following values, depending on your host system capability:

   **D** Direct-Coupled Interlock (DCI) mode

   **S** Maximum 3.0 MBps data streaming speed

   **S4** Maximum 4.5 MBps data streaming speed

   The transfer mode and channel transfer speed specified for the PROTOCL parameter must conform with the PCA setting for transfer mode and channel transfer speed.
3. The UNIT parameter on the CNTLUNIT and IODEVICE statements must be set to 3172.
4. When an ESCON Converter is the channel path, the CHPID TYPE parameter must be set to FX, otherwise it is set to BL.

## Defining the 2216 to the Operating System

The following definitions apply to a 2216 with a channel adapter.

### 2216 Definition for VM/SP

The 2216 must be defined to a VM/SP operating system. This definition is accomplished by updating the real I/O configuration file (DMKRIO) with entries for the 2216 in the RDEVICE and the RCTLUNIT macros. In the following example, 640 is the base unit address and the size of the address range is 32.

```
RDEVICE ADDRESS=(640,32),DEVTYPE=3088
RCTLUNIT ADDRESS=640,CUTYPE=3088,FEATURE=32-DEVICE
```

### 2216 Definition for VM/XA and VM/ESA

The 2216 must be defined to a VM/Extended Architecture (VM/XA or VM/ESA operating system. This definition is accomplished by updating the real I/O configuration file (HCPRIO) with an entry for the 2216 in the RDEVICE macro. In the following examples, 640 and 2A0 are base control unit addresses. The address range size, as defined in the UCW or IOCP, is 8 in both examples.

The following example is a VM/XA HCPRIO definition:

```
RDEVICE ADDRESS=(640,8),DEVTYPE=CTCA
```

The following example is a VM/ESA HCPRIO definition:

```
RDEVICE ADDRESS=(2A0,8),DEVTYPE=CTCA
```

## 2216 Definition for MVS/XA and MVS/ESA without HCD

**Note:** To define a 2216 on an MVS/ESA system with HCD, see "2216 Definition for MVS/ESA with HCD"

The 2216 must be defined to an IBM Multiple Virtual Storage/Extended Architecture (MVS/XA) or MVS/ESA operating system. This definition is accomplished by updating the MVS Control Program with an entry for the 2216 in the IODEVICE macro.

For ESCON channels, an example IODEVICE macro is:

```
IODEVICE UNIT=3172,ADDRESS(540,8)
```

For parallel channels, an example IODEVICE macro is:

```
IODEVICE UNIT=CTC,ADDRESS(640,8)
```

The base control unit addresses are 640 and 540. The address range size, as defined in the UCW or IOCP, is 8 in both examples.

## 2216 Definition for MVS/ESA with HCD

The hardware configuration definition (HCD) component of MVS/ESA SP Version 4.2 and 4.3 with APAR #OY67361 offers an improved method of defining system hardware configuration for 2216. Several complex steps required for entering hardware configuration data can be accomplished using an interactive dialog with HCD.

The required configuration data for the 2216 is:

1. When using HCD, with APAR #OY67361, the 2216 is defined as (UNIT=3172).

   ```
   IODEVICE UNIT=3172,ADDRESS(740,8)
   ```

2. Without HCD, the 2216 is defined for:
   - Parallel channels as a 3088 device (UNIT = 3088 or CTC)

     ```
     IODEVICE UNIT=CTC,ADDRESS(840,8)
     ```

   - ESCON channels as a serial CTC device (UNIT = SCTC)

     ```
     IODEVICE UNIT=SCTC,ADDRESS(A40,8)
     ```

**Notes:**

1. If you are using HCD for MVS Version 4 to define your ESCON host connection, you will need APAR # OY67361 to obtain the UIM support for the device definition (UNIT=3172).
2. When migrating your IOCP definition and operating system definitions to the HCD environment, it is important that all 2216 device statements be changed to device type (UNIT=3172).

## 2216 Definition for VSE/ESA

The 2216 must be defined to a VSE/ESA operating system. This definition is accomplished by supplying an ADD statement for each channel unit address at initial program load (IPL) time. Code the device type on the ADD statement as CTCA,EML as shown in the following example:

```
ADD 640,CTCA,EML
```

The base control unit address is 640 in the example. For the number of channel unit addresses added, increment the IOTAB storage macro by this count.

# Defining the 2216 to Host Programs

The section has configuration definitions with samples of host definitions required to connect to the 2216 channel adapter.

## Configuring the Host for TCP/IP

TCP/IP can connect to the 2216 channel adapter using either LCS or MPC+. When using MPC+, TCP/IP in the host goes through VTAM to the 2216.

**Note:** When referring to MPC+, the host uses the term HPDT MPC while the 2216 uses the term MPC+.

TCP is configured on a host by modifying the TCP/IP profile. The default name for the TCP/IP profile data set is TCPIP.PROFILE.TCPIP for MVS and PROFILE TCPIP for VM. Each channel connection requires:

- A DEVICE statement for each subchannel pair or group.
- A LINK statement for each LCS interface on the 2216. Multiple LINKs can be defined for a single DEVICE.
- A LINK statement for each IP address supported over an MPC+ Group. Only one LINK can be defined for a single DEVICE.
- An entry in the HOME statement for each LINK statement.
- Entries in the GATEWAY statement for the link to be used (if ROUTED is not being used)
- A START command for each device

See the OS/390 TCP/IP OpenEdition publications for more information on configuring TCP/IP.

***Configuring the Host for TCP/IP using LCS:*** These are the statements required to configure TCP/IP on the host when using LCS:

*DEVICE and LINK statements:* The format of the DEVICE and LINK statements are:

```
DEVICE devicename LCS subchannel
LINK   iplinkname LANtype LANnumber devicename
```

where:

**devicename**
is a local name to distinguish devices. You need a START statement for this device name at the end of the TCP/IP profile as shown in “Example of TCP/IP Commands when Using LCS” on page 324.

**LCS subchannel**
is the even subchannel of the two LCS subchannels that this connection to the 2216 will use.

**iplinkname**
is a local name to distinguish links. This name can help you identify which link is being configured.

**LANtype**
is the type of link.

**LANnumber**
is obtained from the 2216 by using the LIST NETS command of the appropriate NETWORK submenu.

*HOME Command:* Specify IP addresses for each channel connection using the following format:

```
HOME hostipadd iplinkname
```

where:

**hostipadd**
is the host's IP address for this connection to the TCP/IP network.

**Note:** This IP address must be a unique address in the same IP subnetwork as the corresponding IP address coded in the 2216.

**iplinkname**
is the link name defined by the LINK statement as described in "DEVICE and LINK statements" on page 322.

*GATEWAY Command:* Specify routing information if you are not using the ROUTED server.

```
GATEWAY network  first hop  driver  packet size  subn mask  subn value
```

where:

**network**
is the IP address for the network. The default value is DEFAULTNET, which specifies a default routing entry for any network not explicitly routed.

**first hop**
Specify one of the following:

An equal sign (=) meaning that messages are routed directly to destinations on that network or directly to that host. This is not supported for DEFAULTNET.

The Internet address of a gateway or router that you can reach directly, and that forwards messages for the destination network or host.

**driver** is the *iplinkname* defined by the LINK statement as described in "DEVICE and LINK statements" on page 322.

**packet size**
is the maximum transmission unit in bytes for the network or host.

**subn mask**
is a bit mask that defines the bits of the host field that make up the subnet field.

**subn value**
is the value of the subnet field.

*START Command:* Start all the interfaces:

```
START devicename
```

where:

**devicename**
is the parameter defined by the DEVICE statement as described in "DEVICE and LINK statements" on page 322.

***Example of TCP/IP Commands when Using LCS:***

```
DEVICE LCS1 LCS   108
LINK   TR1  IBMTR 0 LCS1
HOME
    16.51.136.199 TR1      1
GATEWAY
    16.51.136.201 = TR1 4000  HOST    2
    DEFAULTNET 16.51.136.201 TR1 4000 0
START LCS1
```

1 16.51.136.199 is in the same IP subnetwork as the 2216 LCS interface's IP address.

2 16.51.136.201 is the 2216 LCS interface's IP address.

***Configuring the Host for TCP/IP using MPC+:*** These are the statements required to configure TCP/IP on the host when using MPC+:

*DEVICE and LINK statements:*

```
DEVICE devicename MPCPTP
LINK   iplinkname MPCPTP devicename
```

where:

**devicename**
is the name of the TRL that this connection to the 2216 will use. See "Sample 2216 Definition to TCP/IP for MVS or VM for MPC+" on page 327 for additional information.

**MPCPTP**
specifies an MPC point-to-point link.

**iplinkname**
is a link name to distinguish links. This name can help you identify which link is being configured.

*HOME Command:* Specify IP addresses for each channel connection using the following format:

```
HOME hostipadd iplinkname
```

where:

**hostipadd**
is the host's IP address for this connection to the TCP/IP network.

**Note:** This IP address must be a unique address in the same IP subnetwork as the corresponding IP address coded in the 2216.

**iplinkname**
is the link name defined by the LINK statement as described in "DEVICE and LINK statements".

*GATEWAY Command:* Specify routing information if you are not using the ROUTED server.

```
GATEWAY network  first hop  driver  packet size  subn mask  subn value
```

where:

**network**
is the IP address for the network. The default value is *defaultnet*, which specifies a default routing entry for any network not explicitly routed.

**first hop**
Specify one of the following:

An equal sign (=) meaning that messages are routed directly to destinations on that network or directly to that host. This is not supported for DEFAULTNET.

The Internet address of a gateway or router that you can reach directly, and that forwards messages for the destination network or host.

**driver** is the *iplinkname* defined by the LINK statement as described in "DEVICE and LINK statements" on page 324.

**packet size**
is the maximum transmission unit in bytes for the network or host.

**subn mask**
is a bit mask that defines the bits of the host field that make up the subnet field.

**subn value**
is the value of the subnet field.

*START Command:* Start all the interfaces:

```
START devicename
```

where:

**devicename**
is the parameter defined by the DEVICE statement as described in "DEVICE and LINK statements" on page 324.

***Example of TCP/IP Commands when Using MPC+:*** For MPC+:

```
DEVICE IPTRL1 MPCPTP
LINK   LINK1 MPCPTP IPTRL1
HOME
    198.10.70.199 LINK1    1
GATEWAY
    198.10.70.203 = LINK1 16000 HOST    2
    DEFAULTNET 198.10.70.203 LINK1
START IPTRL1
```

1 198.10.70.199 is in the same IP subnetwork as one of the 2216 MPC+ interface's IP addresses.

2 198.10.70.203 is one of the 2216 MPC+ interface's IP addresses.

## Sample 2216 Definition to TCP/IP for MVS or VM for LCS

The following is an example of TCP/IP definitions that would be provided to the host computer in the TCP/IP Profile data set to configure an LCS device. The default name for the TCP/IP profile data set is TCPIP.PROFILE.TCPIP for MVS and PROFILE TCPIP for VM.

First, 2216 devices and links are defined to TCP/IP.

There is a DEVICE statement for each subchannel pair that is used to access 2216s. The first address specified must be an ***even*** address. In this example, two devices (subchannel pairs) are defined: one at address 640 and one at address 642. These devices could be in the same or different 2216s. A device type of LCS (LAN Channel Station) is used to define these devices to TCP/IP.

There is a LINK statement for each LAN adapter that is accessible from these devices. In this example, one Ethernet/802.3 Adapter is assigned to the device using subchannels 640 and 641, two Token-Ring adapters are assigned to the device using 642 and 643, and one FDDI adapter using 644 and 645. These two Token-Ring adapters are in the same 2216 because they are associated with the same device. The LINK number for each adapter (0 and 1 in this example) is assigned by the 2216 when you add an adapter to a profile.

**Note:** Two subchannel addresses are required for sending and receiving (for example, 640 and 641), but only the first address is defined.

```
DEVICE LCS1 LCS 640
LINK ETH1 ETHEROr802.3  0 LCS1
DEVICE LCS2 LCS 642
LINK TR1  IBMTR     0 LCS2
LINK TR2  IBMTR     1 LCS2
DEVICE LCS3 LCS 644
LINK FD1  FDDI      0 LCS3
```

**Note:** In this example, `0` and `1` are the LAN numbers for these connections.

This section of the example TCP/IP profile defines the local host internet addresses:

```
HOME

193.5.2.1      ETH1
130.50.75.1    TR1
130.50.76.1    TR2
195.10.70.1    FD1
```

This section of the example TCP/IP profile represents the LAN/WAN gateway definition:

```
GATEWAY

Network  First hop Driver  Packet Size Subnet mask Subnet value
193.5.2      =      ETH1    1500           0
130.50       =      TR1     2000        0.0.255.0   0.0.75.0
130.50       =      TR2     2000        0.0.255.0   0.0.76.0
195.10       =      FD1     4000        0.0.255.0   0.0.70.0
```

This section of the example TCP/IP profile activates the LCS devices:

```
START LCS1
START LCS2
START LCS3
```

The following examples illustrate various ways that LAN adapters can be specified and linked to subchannel pairs in the TCP/IP profile.

Two LCS devices for the two subchannel pairs 40,41 and 42,43 and four LAN adapters are defined in the 2216 as follows:

```
DEVICE LCS1 LCS 640
LINK ETH1 ETHERNET 0 LCS1
LINK ETH2 ETHERNET 1 LCS1
DEVICE LCS2 LCS 642
LINK TRN1 IBMTR    0 LCS2
LINK TRN2 IBMTR    1 LCS2
```

Four LCS devices for the four subchannel pairs 40,41; 42,43; 44,45; and 46,47 and four LAN adapters are defined in the 2216 as follows:

```
DEVICE LCS1 LCS 640
LINK ETH1 ETHERNET 0 LCS1
DEVICE LCS2 LCS 642
LINK ETH2 ETHERNET 1 LCS2
DEVICE LCS3 LCS 644
LINK TRN1 IBMTR    0 LCS3
DEVICE LCS4 LCS 646
LINK FD1  FDDI      0 LCS4
```

One LCS device for the subchannel pair 40,41 and four LAN adapters are defined in the 2216 as follows:

```
DEVICE LCS1 LCS 640
LINK ETH1 ETHERNET 0 LCS1
LINK ETH2 ETHERNET 1 LCS1
LINK ETH3 ETHERNET 2 LCS1
LINK ETH4 ETHERNET 3 LCS1
```

## Sample 2216 Definition to TCP/IP for MVS or VM for MPC+

The following is an example of TCP/IP definitions that would be provided to the host computer in the TCP/IP Profile data set to configure an MPC+ device. The default name for the TCP/IP profile data set is TCPIP.PROFILE.TCPIP for MVS and PROFILE TCPIP for VM.

First, in the VTAM host, define a TRL over which TCP/IP should run.

```
IPTRL VBUILD TYPE=TRL
IPTRL1 TRLE LNCTL=MPC,
       MAXBFRU=6,
       READ=(06),
       WRITE=(07)
       REPLYTO=3.0
```

**Notes:**

1. Multiple TCP/IP stacks can use the same TRL.
2. See “Configuring the VTAM Host for MPC+ for IP” on page 334 for more details.

There is a DEVICE statement for each TRL that is to be used by TCP/IP.

```
DEVICE IPTRL1 MPCPTP
```

There is a LINK statement for each TCP/IP local host internet address using the TRL. For a given TCP/IP stack, there can only be one LINK statement per TRL.

```
LINK LINK1 MPCPTP IPTRL1
```

This section of the example TCP/IP profile defines the local host internet addresses and associates the addresses with a link:

```
HOME

198.10.70.199 LINK1 1
```

1 198.10.70.199 is in the same IP subnetwork as one of the 2216 MPC+ interface’s IP addresses.

This section of the example TCP/IP profile represents the LAN/WAN gateway definition:

```
GATEWAY

Network        First hop      Driver  Packet Size  Subnet mask  Subnet value
198.10.70.203    =            LINK1   16000        HOST   1
DEFAULTNET     198.10.70.203  LINK1   16000        0
```

1 198.10.70.203 is one of the 2216 MPC+ interface’s IP addresses.

This section of the example TCP/IP profile activates the MPC+ device:

```
START IPTRL1
```

**Note:** Prior to activating the device, activate the associated TRL if it is not already active.

## Configuring the Host for HPDT UDP:

**Important!**
Before you configure your S/390 host for HPDT UPD, you ***must*** have APAR OW31305 installed.

HPDT UDP can connect to the 2216 only through an ESCON Channel Adapter using MPC+.

**Notes:**

1. The 2216 refers to HPDT UDP as UDP+.
2. UDP+ is not supported on a Parallel Channel Adapter (PCA).

HPDT UDP is configured and run using OS/390 TCP/IP OpenEdition (OE) on the host. Therefore, OS/390 TCP/IP OpenEdition must be installed on the host.

HPDT UDP commands are used to configure and control HPDT UDP resources. See *OS/390 TCP/IP Update Guide* for details on installing and using OE for HPDT UDP.

To configure and activate an HPDT UDP connection on the 2216, issue the **oeifconfig** command, shown in the following example.

```
oeifconfig interface_name source_IP_address destination_IP address mtu nnnn
```

where:

**interface_name**
is the name of the TRL for the HPDT UDP connection.

This will cause the TRL to be activated if it is not already active.

**Note:** See “Configuring the VTAM Host for MPC+ for IP” on page 334 for information about TRLs.

**source_IP_address**
is the local IP address in the host for HPDT UDP connection.

This local IP address must be in the same IP subnetwork as the IP address coded for the UDP+ MPC+ network handler in the 2216.

**destination_IP_address**
is the destination IP address in the 2216 for this HPDT UDP connection.

This destination IP address must equal the IP address coded on the UDP+ MPC+ network handler in the 2216.

**mtu nnnn**
where `nnnn` is the maximum transmission unit size for the HPDT UDP connection.

This mtu size must be equal to Maxdata coded on the UDP+ MPC+ network handler in the 2216. If the values are not equal, then the HPDT UDP connection will not come up.

**Note:** Because this is a point-to-point connection, there is no need to code the netmask parameter on this command.

## Sample 2216 Definition to HPDT UDP for MVS or VM

First, in the VTAM host, define a TRL over which HPDT UDP should run.

```
UDPTRL VBUILD TYPE=TRL
TRL1   TRLE LNCTL=MPC,
       MAXBFRU=9,
       READ=(EA0),
       WRITE=(EA1),
       REPLYTO=3.0
```

**Note:** See “Configuring the VTAM Host for MPC+ for IP” on page 334 for more details.

Next, go to the OE environment and issue the following command:

```
oeifconfig trl1 198.10.60.199 198.10.60.203 mtu 16384
```

## VTAM Control Blocks Used to Configure LSA at the Host

Configuring the VM or MVS host requires entries in two VTAM control blocks:

- External communication adapter (XCA) major node definition file
- Switched major node configuration file

For more information on configuring VTAM, refer to *VTAM Resource Definition Reference*.

***XCA Major Node Definition File - Sample:*** Defining an XCA major node requires coding VTAM definition statements to define the following characteristics:

- Node type (VBUILD definition statement)
- Port used by the LAN (PORT definition statement)
- Switched lines attached through the 2216 channel adapter (GROUP, LINE, and PU definition statements)

You must code a VBUILD definition statement and a PORT definition statement for the major node, and code GROUP, LINE, and PU definition statements for minor nodes.

You must also assign service access points (SAPs) to be used for each virtual channel to a LAN or emulated LAN.

***Switched Major Node Definition File - Sample:*** The switched major node definition file defines the workstations that VTAM will be able to access through the 2216 channel adapter, and identifies:

- Node type (VBUILD definition statement)
- Network Resources (PU and LU definition statements)

To define the 2216 LAN/WAN gateway to VTAM, the appropriate LAN adapter in the IBM 2216 must be associated with a subchannel address. This association is defined to VTAM in a major node definition that is supported by VTAM Version 3 Release 4 and VTAM Version 4 Release 1.

## Configuring an LSA Direct Connection at the VTAM Host

Configuring the VM or MVS host requires entries in two VTAM control blocks, the XCA Major Node Definition File and the Switched Major Node Definition File. See “VTAM Control Blocks Used to Configure LSA at the Host” for a description of the purpose of these control blocks and references to VTAM publications

***XCA Major Node Definition File - Sample:***

```
ROUTE6B1 VBUILD TYPE=XCA
PORT6B1  PORT   CUADDR=0CB,ADAPNO=0,TIMER=60,SAPADDR=08,              C
               MEDIUM=RING
GRP6B1   GROUP  DIAL=YES
************************
LN06B001 LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU06B001 PU    ISTATUS=ACTIVE
```

**Notes:**

1. ADAPNO is the LAN number for the 2216 LSA interface.
2. CUADDR is the channel address. This corresponds to the Device Address (three hexadecimal characters defining the channel address) for the 2216 LSA interface.
3. MEDIUM=RING for Token Ring, MEDIUM=CSMACD for Ethernet, and MEDIUM=FDDI for FDDI. This corresponds to the value specified for LANtype for the 2216 interface.

***Switched Major Node Definition File - Sample:***

```
PS06SW  VBUILD TYPE=SWNET
PS06PU  PU ADDR=01,IDBLK=05D,IDNUM=54445,MAXOUT=7,PACING=0,VPACING=0,  C
               SSCPFM=USSSCS,MAXDATA=4105,MODETAB=LMT3270,MAXPATH=1,   C
               ANS=CONT,ISTATUS=ACTIVE,DLOGMOD=B22NNE
PS06LU2 LU LOCADDR=02
PS06LU3 LU LOCADDR=03
PS06LU4 LU LOCADDR=04
PS06LU5 LU LOCADDR=05
```

## Configuring an LSA APPN Connection at the VTAM host

Configuring the VM or MVS host requires entries in two VTAM control blocks, the XCA Major Node Definition File and the Switched Major Node Definition File. See “VTAM Control Blocks Used to Configure LSA at the Host” on page 329 for a description of the purpose of these control blocks and references to VTAM publications.

***XCA Major Node Definition File - Sample:***

```
P15AP63X VBUILD TYPE=XCA
PORT63X  PORT   CUADDR=0CD,ADAPNO=0,TIMER=60,SAPADDR=04,              C
               MEDIUM=CSMACD
GRP63X   GROUP  DIAL=YES
************************
LN630403 LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU630403 PU    ISTATUS=ACTIVE
```

**Notes:**

1. ADAPNO is the LAN number for the 2216 LSA interface.
2. CUADDR is the channel address. This corresponds to the Device Address (two hexadecimal characters defining the lower byte of the channel address) for the 2216 interface.
3. MEDIUM=RING for Token Ring and MEDIUM=CSMACD for Ethernet. This corresponds to the value specified for LANtype for the 2216 LSA interface.

***Switched Major Node Definition File - Sample:***

```
LS601    VBUILD TYPE=SWNET
CS601   PU ADDR=02,CPNAME=C210,MAXOUT=7,PACING=0,VPACING=0,            C
               CPCP=YES,MAXDATA=4105,MODETAB=LMT3270,MAXPATH=10,       C
               CONNTYPE=APPN,DYNLU=YES
```

## Configuring an LSA DLSw Connection at the VTAM Host

Configuring the VM or MVS host requires entries in two VTAM control blocks, the XCA Major Node Definition File and the Switched Major Node Definition File. See “VTAM Control Blocks Used to Configure LSA at the Host” on page 329 for a description of the purpose of these control blocks and references to VTAM publications.

### *XCA Major Node Definition File - Sample:*

```
P15AP60X VBUILD TYPE=XCA
PORT60X  PORT   CUADDR=0CC,ADAPNO=1,TIMER=60,SAPADDR=04,              C
               MEDIUM=CSMACD
GRP60X   GROUP  DIAL=YES
************************
LN600403 LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU600403 PU    ISTATUS=ACTIVE
LN600404 LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU600404 PU    ISTATUS=ACTIVE
```

**Notes:**

1. ADAPNO is the LAN number for the 2216 LSA interface.
2. CUADDR is the channel address. This corresponds to the Device Address (two hexadecimal characters defining the lower byte of the channel address) for the 2216 interface.
3. MEDIUM=RING for Token Ring and MEDIUM=CSMACD for Ethernet. This corresponds to the value specified for LANtype for the 2216 LSA interface.

### *Switched Major Node Definition File - Sample:*

```
PSK5SW  VBUILD TYPE=SWNET
PSK5PU  PU ADDR=03,IDBLK=05D,IDNUM=07251,MAXOUT=7,PACING=0,VPACING=0,  C
               DLOGMOD=B22NNE,                                         C
               SSCPFM=USSSCS,MAXDATA=2000,MODETAB=LMT3270
PSK5LU2 LU LOCADDR=02
PSK5LU3 LU LOCADDR=03
PSK5LU4 LU LOCADDR=04
PSK5LU5 LU LOCADDR=05
PSK5LU6 LU LOCADDR=06
```

## Configuring an LSA DLSw Local Conversion at the VTAM Host

Configuring the VM or MVS host requires entries in two VTAM control blocks, the XCA Major Node Definition File and the Switched Major Node Definition File. See “VTAM Control Blocks Used to Configure LSA at the Host” on page 329 for a description of the purpose of these control blocks and references to VTAM publications.

### *XCA Major Node Definition File - Sample:*

```
P15AP60X VBUILD TYPE=XCA
PORT60X  PORT   CUADDR=0CC,ADAPNO=1,TIMER=60,SAPADDR=04,              C
               MEDIUM=CSMACD
GRP60X   GROUP  DIAL=YES
************************
LN600403 LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU600403 PU    ISTATUS=ACTIVE
LN600404 LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU600404 PU    ISTATUS=ACTIVE
```

**Notes:**

1. ADAPNO is the LAN number for the 2216 LSA interface.
2. CUADDR is the channel address. This corresponds to the Device Address (two hexadecimal characters defining the lower byte of the channel address) for the 2216 interface.

3. MEDIUM=RING for Token Ring and MEDIUM=CSMACD for Ethernet. This corresponds to the value specified for LANtype for the 2216 LSA interface.

***Switched Major Node Definition File - Sample:***

```
PS06SW  VBUILD TYPE=SWNET,MAXDLUR=20
PS06PU  PU ADDR=01,IDBLK=05D,IDNUM=54445,MAXOUT=7,PACING=0,VPACING=0,  C
               SSCPFM=USSSCS,MAXDATA=4105,MODETAB=LMT3270,MAXPATH=1,   C
               ANS=CONT,ISTATUS=ACTIVE,DLOGMOD=B22NNE
PS06LU2 LU LOCADDR=02
PS06LU3 LU LOCADDR=03
PS06LU4 LU LOCADDR=04
PS06LU5 LU LOCADDR=05

PSK5SW  VBUILD TYPE=SWNET
PSK5PU  PU ADDR=03,IDBLK=05D,IDNUM=07251,MAXOUT=7,PACING=0,VPACING=0,  C
               DLOGMOD=B22NNE,                                         C
               SSCPFM=USSSCS,MAXDATA=2000,MODETAB=LMT3270
PSK5LU2 LU LOCADDR=02
PSK5LU3 LU LOCADDR=03
PSK5LU4 LU LOCADDR=04
PSK5LU5 LU LOCADDR=05
PSK5LU6 LU LOCADDR=06
```

The following examples show XCA and SWNET macros that define the LAN major node for a Token-Ring adapter and an Ethernet adapter, respectively. In the examples:

- GROUP1T, GROUP1E, and GROUP1F represent resources connected to the LAN that require a VBUILD TYPE=SWNET.
- GROUP2T, GROUP2E, and GROUP2F represent a connection for the PU 5 node.

The mode table and default mode entries are examples only. Be sure to use the mode tables and mode entries defined in your installation.

```
TRLAN1   VBUILD TYPE=XCA
PORT1    PORT   MEDIUM=RING,ADAPNO=0,CUADDR=644,TIMER=60,SAPADDR=4
GROUP1T  GROUP  DIAL=YES    * Switched Attachment
LINE1TA  LINE   ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU1TA    PU     ISTATUS=ACTIVE
LINE1TB  LINE   ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU1TB    PU     ISTATUS=ACTIVE
GROUP2T  GROUP  DIAL=NO     * Leased Definition
LINE2T   LINE   USER=SNA    * Multi-domain Connection
PU2T     PU     MACADDR=400000000001,TGN=1,SUBAREA=2,SAPADDR=4,PUTYPE=5


ENLAN2   VBUILD TYPE=XCA
PORT2    PORT   MEDIUM=CSMACD,ADAPNO=0,CUADDR=645,TIMER=60,SAPADDR=4
GROUP1E  GROUP  DIAL=YES    * Switched Attachment
LINE1EA  LINE   ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU1EA    PU     ISTATUS=ACTIVE
LINE1EB  LINE   ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
PU1EB    PU     ISTATUS=ACTIVE
GROUP2E  GROUP  DIAL=NO     * Leased Definition
LINE2E   LINE   USER=SNA    * Multi-domain Connection
PU2E     PU     MACADDR=400000000002,TGN=2,SUBAREA=2,SAPADDR=4,PUTYPE=5
```

The following examples are the switched major node definitions.

```
LS100SW VBUILD TYPE=SWNET,MAXGRP=400,MAXNO=400
CS100001 PU  ADDR=01,PUTYPE=2,MAXPATH=4,ANS=CONT,DLOGMOD=B22NNE,
         ISTATUS=ACTIVE,MAXDATA=521,IRETRY=YES,MAXOUT=7,
         PASSLIM=5,IDBLK=111,IDNUM=00001,MODETAB=LMT3270
         PATH  DIALNO=0104400000000004,GRPNM=GROUP1T
S00102   LU   LOCADDR=2
CS100002 PU  ADDR=02,PUTYPE=2,MAXPATH=4,ANS=CONT,DLOGMOD=B22NNE,
         ISTATUS=ACTIVE,MAXDATA=521,IRETRY=YES,MAXOUT=7,
         PASSLIM=5,CPNAME=MYNS2,MODETAB=LMT3270
         PATH  DIALNO=0104400000000005,GRPNM=GROUP1T
S00200  LU   LOCADDR=0,DLOGMOD=LU62MODE
S00202  LU   LOCADDR=2
```

```
CS100003 PU  ADDR=03,PUTYPE=2,MAXPATH=4,ANS=CONT,DLOGMOD=B22NNE,
         ISTATUS=ACTIVE,MAXDATA=521,IRETRY=YES,MAXOUT=7,
         PASSLIM=5,IDBLK=111,IDNUM=00003,MODETAB=LMT3270
         PATH  DIALNO=0104400000000006,GRPNM=GROUP1E
S00302   LU   LOCADDR=2
CS100004 PU  ADDR=04,PUTYPE=2,MAXPATH=4,ANS=CONT,DLOGMOD=B22NNE,
         ISTATUS=ACTIVE,MAXDATA=521,IRETRY=YES,MAXOUT=7,
         PASSLIM=5,IDBLK=111,IDNUM=00004,MODETAB=LMT3270
         PATH  DIALNO=0104400000000007,GRPNM=GROUP1E
S00402  LU   LOCADDR=2

CS100005 PU  ADDR=05,PUTYPE=2,MAXPATH=4,ANS=CONT,DLOGMOD=B22NNE,
         ISTATUS=ACTIVE,MAXDATA=521,IRETRY=YES,MAXOUT=7,
         PASSLIM=5,IDBLK=111,IDNUM=00005,MODETAB=LMT3270
         PATH  DIALNO=0104400000000008,GRPNM=GROUP1F
S00502   LU   LOCADDR=2
CS100006 PU  ADDR=06,PUTYPE=2,MAXPATH=4,ANS=CONT,DLOGMOD=B22NNE,
         ISTATUS=ACTIVE,MAXDATA=521,IRETRY=YES,MAXOUT=7,
         PASSLIM=5,IDBLK=111,IDNUM=00006,MODETAB=LMT3270
         PATH  DIALNO=0104400000000005,GRPNM=GROUP1F
S00602  LU   LOCADDR=2
```

For more information about VTAM definitions, see:

*VTAM V4R4 Network Implementation Guide,* SC31-8370

*VTAM V4R4 Resource Definition Reference,* SC31-8377

## Configuring the VTAM Host for MPC+ for APPN

Configuring the VTAM host for MPC+ for APPN requires entries in two VTAM control blocks, the Local SNA Major Node and the Transport Resource List (TRL) Major Node, and a change to the VTAM start-up parameters. A TRL corresponds to an MPC+ Group. For more information on configuring VTAM, refer to *VTAM Resource Definition Reference*.

***Local SNA Major Node:*** Use the following definition statements to configure a local SNA major node in VTAM:

```
UTYLSNA VBUILD TYPE=LOCAL
UTYHCC1 PU     TRLE=UHCC1,XID=YES,CONNTYPE=APPN,CPCP=YES,HPR=YES
```

***Transport Resource List (TRL) Major Node:***

```
BC4UTRL VBUILD TYPE=TRL
UHCC1   TRLE LNCTL=MPC,                                          C
             MAXBFRU=n,                                          C
             READ=(xxx1,xxx2,...),                               C
             WRITE=(yyy1,yyy2,...),                              C
             REPLYTO=3.0
```

where:

**n** is the number of 4K buffer pages VTAM uses to receive data over the channel.

**xxx1,xxx2,...**
are the read subchannel numbers.[1]

**yyy1,yyy2,...**
are the write subchannel numbers.[1]

[1] The subchannels do not need to be contiguous.

The read and write subchannel numbers must be attached to those configured on the 2216 (for example, by way of a message operator mount or an attach command).

**Note:** A “read” subchannel to VTAM is a “write” subchannel to the 2216 and a “write” subchannel to VTAM is a “read” subchannel to the 2216.

***VTAM Start-up Parameters:*** In the VTAM initialization file ATCSTRxx, where xx is defined by the user, define a network node:

```
NODETYPE=NN
```

Since high-performance routing (HPR) is being used, you also should add to this file:

```
HPR=YES
```

**Note:** Only APPN HPR is supported across the MPC+ interface. APPN ISR is not supported.

To activate the APPN PU, activate the associated TRLE (if it is not already active) and then activate the PU.

See *VTAM Operation* for information about VTAM Commands.

## Configuring the VTAM Host for MPC+ for IP

**Note:** UDP+ is not supported on a Parallel Channel Adapter (PCA).

Configuring the VTAM host for MPC+ for TCP/IP or for HPDT UDP requires an entry in the VTAM Transport Resource List (TRL) Major Node control block. A TRL corresponds to an MPC+ Group. For more information on configuring VTAM, refer to *VTAM Resource Definition Reference*.

***Transport Resource List (TRL) Major Node:***

```
TRL    VBUILD TYPE=TRL
TRL1   TRLE LNCTL=MPC,                                    C
       MAXBFRU=n,                                         C
       READ=(xxx1,xxx2,...),                              C
       WRITE=(yyy1,yyy2,...),                             C
       REPLYTO=3.0
```

where:

**n** is the number of 4K buffer pages VTAM uses to receive data over the channel.

**Note:** For both TCP/IP and HPDT UDP, MAXBFRU*4K must be greater than Maxdata coded on the MPC+ network handler in the 2216.

See *OS/390 TCP/IP OpenEdition* and *OS/390 Update Guide* for additional restrictions regarding MAXBFRU.

**xxx1,xxx2,...**
are the read subchannel numbers.[1]

**yyy1,yyy2,...**
are the write subchannel numbers. [1]

[1] The subchannels do not need to be contiguous.

The read and write subchannel numbers must match those configured on the 2216 (for example, by way of a message operator mount or an attach command).

**Note:** A “read” subchannel to VTAM is a “write” subchannel to the 2216 and a “write” subchannel to VTAM is a “read” subchannel to the 2216.

## Planning for 2216 Support

This section describes considerations for support of the 2216 and channel adapter in a network. Analyzing and resolving a communications problem between a channel adapter and a host or LAN may require you to initiate problem resolution procedures for the 2216.

The problem can be:

- A configuration or host definition problem
- A 2216 hardware or software problem
- A channel problem between the 2216 and the host
- A host program, resource, or hardware problem
- A LAN adapter, access unit, or other hardware malfunction
- A LAN workstation program or resource problem
- A communication problem between the 2216 and the LAN
- A problem using hardware or software

Indications of problems come from user reports, or indicators and displayed codes on the 2216 and other devices, or messages displayed by programs. These indications help you determine whether the problem is a hardware, software, or user problem. They also help you isolate the location (2216, LAN, host) and component (device, adapter, channel, or program) of the network that has the problem.

Determining the nature of the problem often indicates which procedures, tools, or additional information may be needed for resolution. The same tools, procedures, and information can also indicate the need to call for service. Problem resolution and service interfaces (NetView, SNMP Client) are available to the customer.

## 2216 Channel Adapter Problem Analysis and Resolution

The 2216 ESCON channel adapter and PCA problem isolation procedures described in *IBM 2216 Nways Multiaccess Connector Service and Maintenance Manual*, SY27–0350 should help you correct the problem, if possible, and determine when to call for service.

## Reconfiguration

Whenever your network grows, shrinks, or rearranges, you may need to reconfigure host programs and 2216 profiles to:

- Balance network traffic and workload
- Migrate to new versions or releases of host programs
- Migrate to new versions or releases of the 2216
- Change host SYSGENs

## Channel Adapter Overview

The ESCON Channel Adapter and PCA provides the 2216 with access to SNA and TCP/IP host applications from LANs, WANs, and ATM.

Figure 23 shows a 2216 connected to a VTAM host through a channel adapter. Each channel adapter provides up to 32 subchannels and up to 16 associated virtual network handlers that can support LAN Channel Station (LCS), Link Services Architecture (LSA), and Multi-Path Channel (MPC+) protocols. Each 2216 can contain up to four channel adapters.

Each ESCON Channel Adapter can provide connections to up to:

- 16 hosts or logical host images using LCS or MPC+ (when used with an ESCON Director)
- 32 hosts or logical host images using LSA (when used with an ESCON Director)
- 15 logical host images in EMIF-capable processors in logically partition mode when no ESCON director is present.

Each PCA can provide connection to one host or one logical host image.

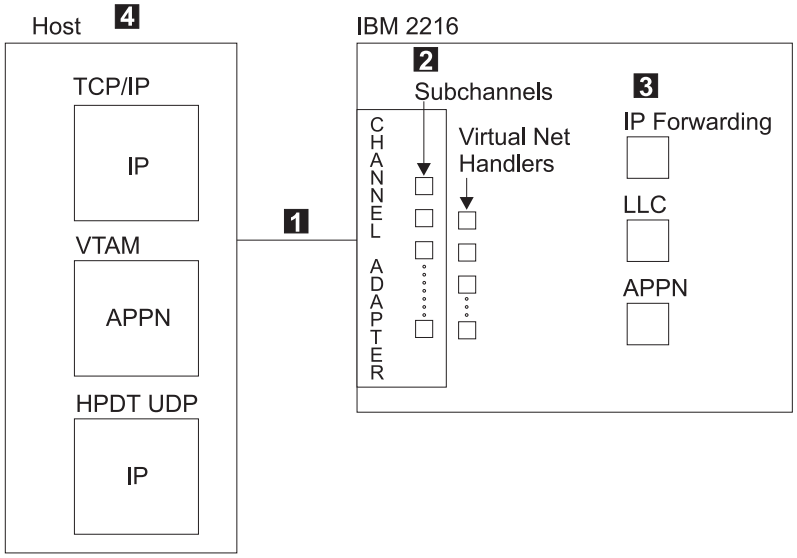



*Figure 23. 2216 Connected to a Host through an ESCON/PCA Channel Adapter - Logical View*

**1** At the physical level, the ESCON Channel Adapter provides a flexible fiber optic connection to communication channels at the host processor. The physical level of the PCA is considerably more complex, requiring a pair of very large copper cables and a set of one to three (depending on configuration) smaller twisted pair copper cables to provide final connection to the PCA.

**2** At the logical level, the channel adapter provides up to 32 *subchannels* and up to 16 associated *virtual network handlers*.

Each virtual network handler supports one of the following protocols:

**LCS** LAN Channel Station

**LSA** Link Services Architecture

**MPC+** Multi-Path Channel+

For each LCS virtual network handler, you must define two subchannels, one for read and one for write; you can define up to 16 LCS virtual network handlers for each channel adapter.

For each LSA virtual network handler, you must define at least one subchannel up to a maximum of 32 subchannels. You can define up to 16 LSA virtual network handlers for each channel adapter.

For MPC+, you can define up to 32 subchannels. You must have at least one read subchannel and at least one write subchannel. You can define up to 16 MPC+ virtual network handlers for each channel adapter.

**Notes:**

1. You can configure LCS and LSA on the same channel adapter.
2. For migration purposes, MPC+ may be configured on the same channel adapter as LCS/LSA. This is not recommended as a long term solution. MPC+ combined with another type of virtual interface (LCS/LSA) on the same adapter could impact the performance benefits provided by the MPC+ interface.

3 The 2216 channel adapter provides services for IP Forwarding, Logical Link Control (LLC), and Advanced Peer-to-Peer Networking (APPN).

4 The virtual net handlers provide connections for transmitting and receiving packets of information for host applications as shown in Figure 24 on page 338 and Figure 25 on page 338.

Once the channel adapter is installed and configured for LCS, LSA, and MPC+, it can provide:

- Hierarchical SNA, including DLSw traffic, and APPN ISR and HPR traffic running over LSA connections. (DLSw and APPN require LLC loopback.)
- TCP/IP traffic running over LCS and MPC+.
- APPN HPR traffic running over MPC+.
- HPDT UDP traffic running over MPC+.

**Note:** UDP+ is not supported on a Parallel Channel Adapter (PCA).

Figure 24 on page 338 shows the basic flow for a channel adapter with LCS and LSA configured, and Figure 25 on page 338 shows the basic flow for a channel adapter for which MPC+ is configured.

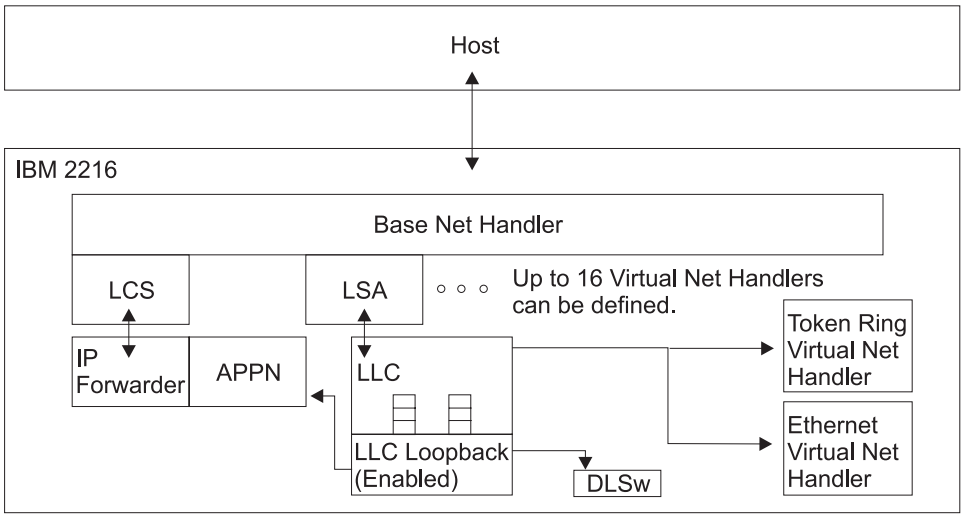



*Figure 24. 2216 Virtual Net Handlers for LCS and LSA*

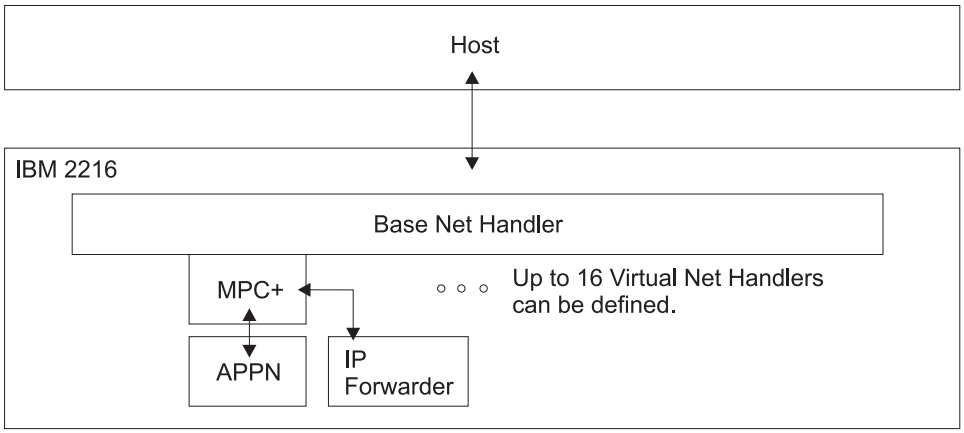



*Figure 25. 2216 Virtual Net Handlers for MPC+*

## LAN Channel Station (LCS) Support

Figure 26 on page 339 shows how TCP/IP data flows from the host, through LCS and other 2216 software components, and out to the LANs/WANs.

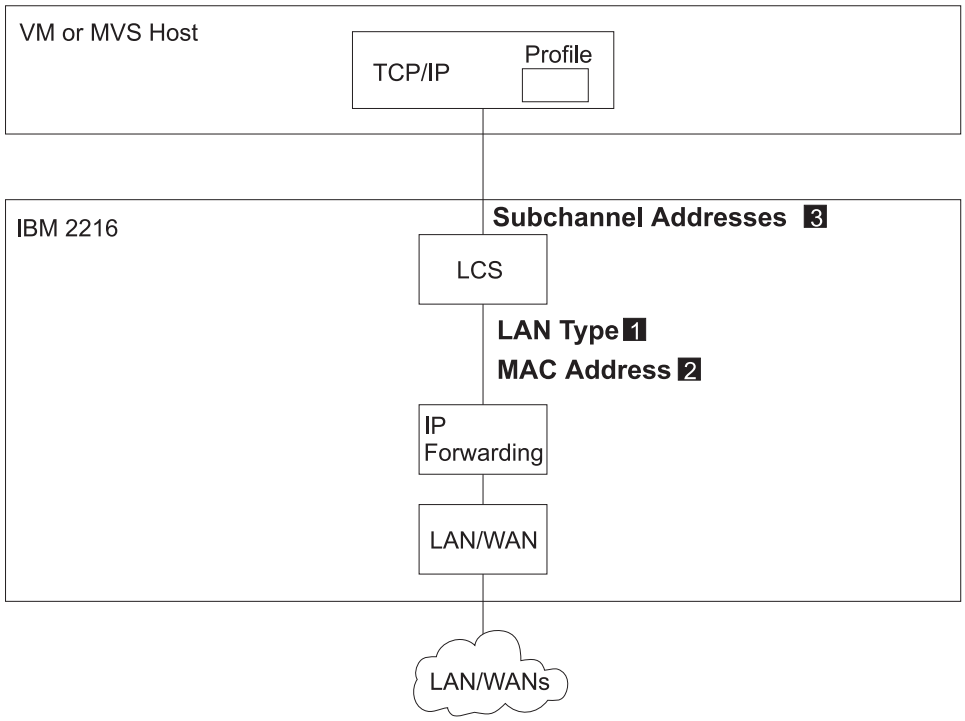



*Figure 26. Configuring LAN Channel Station (LCS) Virtual Net Handlers. This figure shows LCS flow and highlights key parameters at the host and in the 2216.*

## Configuring the 2216 for LCS

Figure 26 shows an LCS connection. To configure the 2216 for LCS:

1. Configure the LAN type (1), either Ethernet, Token Ring, or FDDI, for the connection. This is the frame type that the host expects to send and receive.
2. Configure a unique MAC address (2) for this virtual interface.

   **Note:** If the LAN type is Ethernet, then the MAC address must be in canonical format.
3. Configure the subchannel pair (3) used by this connection as described in "Configuring an LCS Subchannel" on page 358.

You must configure an IP address and mask. Refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*.

There are optional parameters:

**maxdata**
: Maximum size of data handled by this virtual network.

**acklen**
: The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
: The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

For information on the corresponding host definitions, see "Configuring the Host for TCP/IP" on page 322.

# Link Services Architecture (LSA) Support

Link Services Architecture (LSA) permits the VTAM host to communicate with the channel adapter in the 2216.

Figure 27 shows the four types of LSA connections. Their configuration is described in:

- “Configuring an LSA Direct Connection at the 2216” on page 341
- “Configuring an LSA APPN Connection at the 2216” on page 342
- “Configuring an LSA DLSw Connection at the 2216” on page 343
- “Configuring an LSA DLSw Local Conversion at the 2216” on page 345

For information on the corresponding host definitions, see “VTAM Control Blocks Used to Configure LSA at the Host” on page 329.

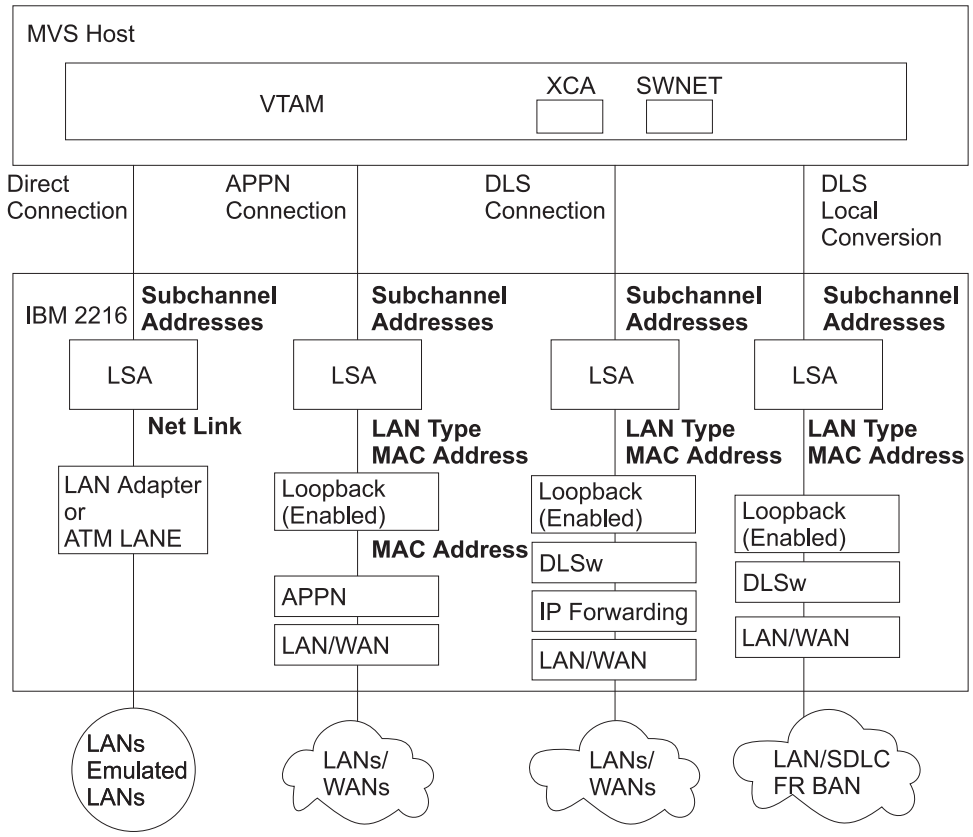



*Figure 27. Configuring Link Services Architecture (LSA) Virtual Net Handlers*

## Configuring an LSA Direct Connection at the 2216

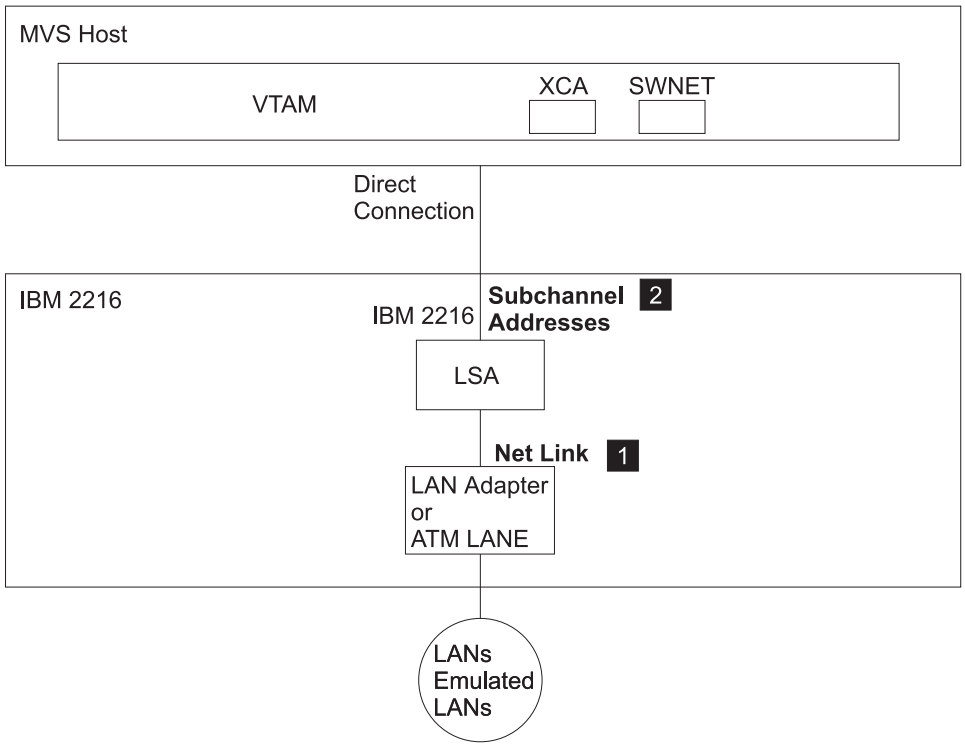



*Figure 28. Configuring Virtual Net Handlers for LSA Direct Connection*

Figure 28 shows an LSA direct connection. To configure the LSA connection:

1. Configure the net link (1). This is the network interface number of the LAN adapter to which the LSA network is linked. This is the interface used by the 2216 to transmit data from the host to the network.
2. Configure the subchannel or subchannels (2 ) used by this connection as described in “Configuring an LSA Subchannel” on page 362.

There are optional parameters:

**maxdata**
Maximum size of data handled by this virtual network.

**acklen**
The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

**Note:** The LSA net will read its MAC address from the 2216 interface configured with the Net Link command.

For information on the corresponding host definitions, see “Configuring an LSA Direct Connection at the VTAM Host” on page 329.

## Configuring an LSA APPN Connection at the 2216

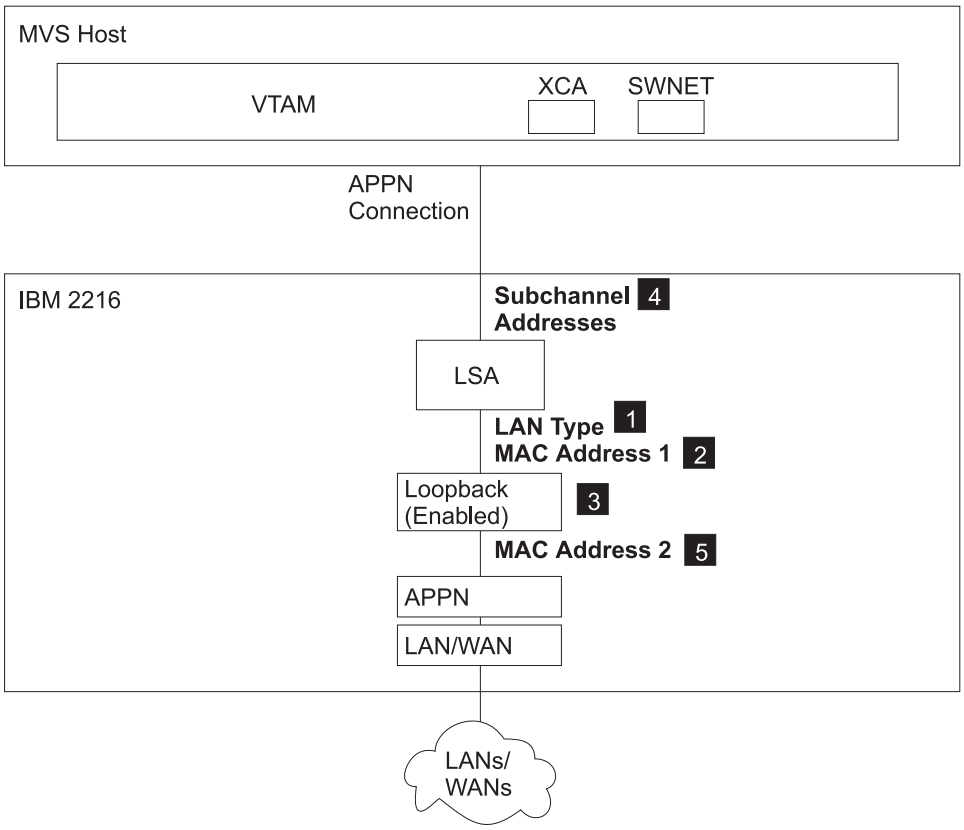



*Figure 29. Configuring Virtual Net Handlers for LSA APPN Connection*

Figure 29 shows an LSA APPN connection. To configure the 2216 for an LSA APPN connection:

1. Enable LSA loopback (3) using the **enable** command.
2. Configure the LAN type (1), either Ethernet or Token Ring

   **Note:** You must configure the same LAN type for both the LSA net and the Loopback net.
3. Configure a unique MAC address (2) to identify the host (VTAM) end of the loopback connection.

   **Note:** If the LAN type is Ethernet, then the MAC address must be in canonical format.
4. Configure the subchannel or subchannels (4) used by this connection as described in “Configuring an LSA Subchannel” on page 362.

   There are optional parameters:

   **maxdata**
   Maximum size of data handled by this virtual network.

   **acklen**
   The size (in bytes) of acknowledgment frames over this interface.

   **blktimer**
   The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

5. Configure APPN to use the APPN loopback net. The APPN port must be configured on the APPN loopback net. To then configure an APPN link station over this APPN port, the destination MAC address of the link station definition should be that of the LSA net.
6. Configure MAC Address 2 (5), a unique MAC address to identify the 2216 (APPN) end of the loopback connection.

   **Note:** If the LAN type is Ethernet, then the MAC address must be in canonical format.

For information on the corresponding host definitions, see “Configuring an LSA APPN Connection at the VTAM host” on page 330.

## Configuring an LSA DLSw Connection at the 2216

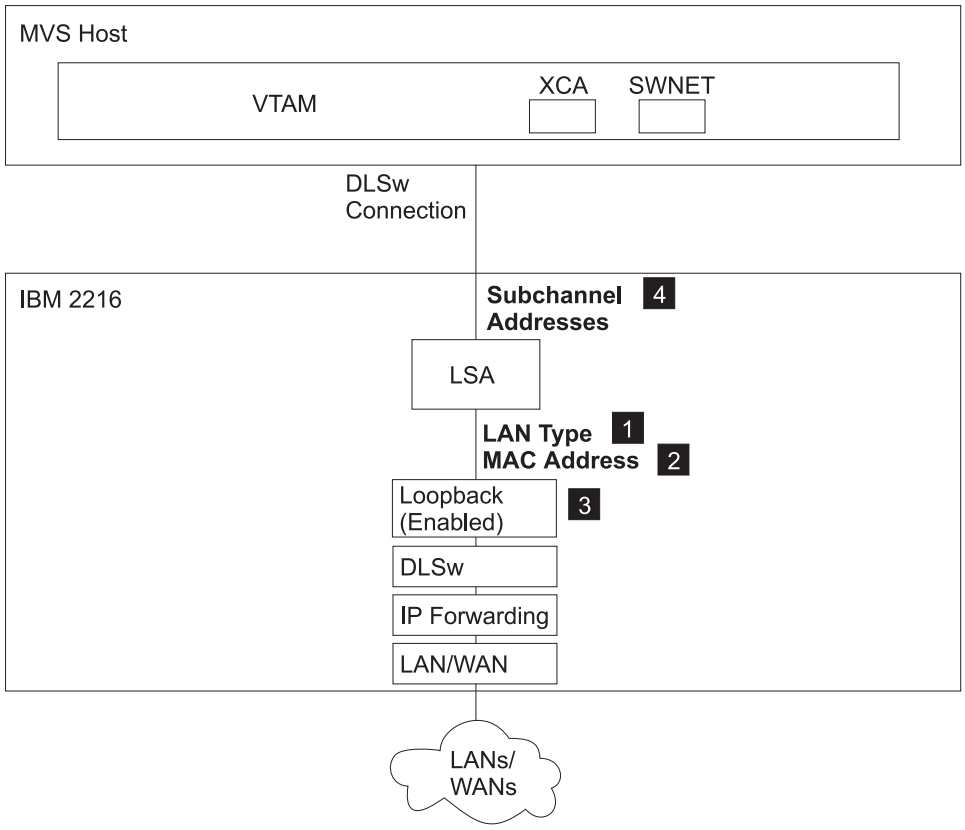



*Figure 30. Configuring Virtual Net Handlers for LSA DLSw Connection*

Figure 30 shows an LSA DLSw connection. To configure an LSA DLSw connection:

1. Enable LSA loopback (3) using the **enable** command.
2. Configure the LAN type (1), either Ethernet or Token Ring. This is the frame type that the host expects to send and receive.
3. Configure a unique MAC address (2) to identify the host (VTAM) end of the loopback connection.

   **Note:** If the LAN type is Ethernet, then the MAC address must be in canonical format.

4. Configure the subchannel or subchannels (4) used by this connection as described in “Configuring an LSA Subchannel” on page 362.
5. Configure DLSw. Configuring DLSw involves enabling DLSw, setting the DLSw segment number, adding the remote DLSw TCP partner and opening the service access points (SAPs) associated with the loopback interface that will be used for DLSw. Configure DLSw from the `config>` prompt.

   Enable DLSw, using the `enable dls` command.

   Set the DLSw segment number using the `set srb` command. The DLSw segment number must be unique.

   Add the remote DLSw TCP partner using the `add tcp` command.

   Open the SAPs that will be used with the LSA loopback interface using the `open` command. The `open` command will prompt for an interface number. Input the interface number assigned to the LSA loopback interface that is defined for use with DLSw.

   For a description of the DLSw configuration parameters, refer to the chapter entitled ″Using and Configuring DLSw″ in the *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1* .

There are optional parameters:

**maxdata**
: Maximum size of data handled by this virtual network.

**acklen**
: The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
: The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

For information on the corresponding host definitions, see “Configuring an LSA DLSw Connection at the VTAM Host” on page 331.

## Configuring an LSA DLSw Local Conversion at the 2216

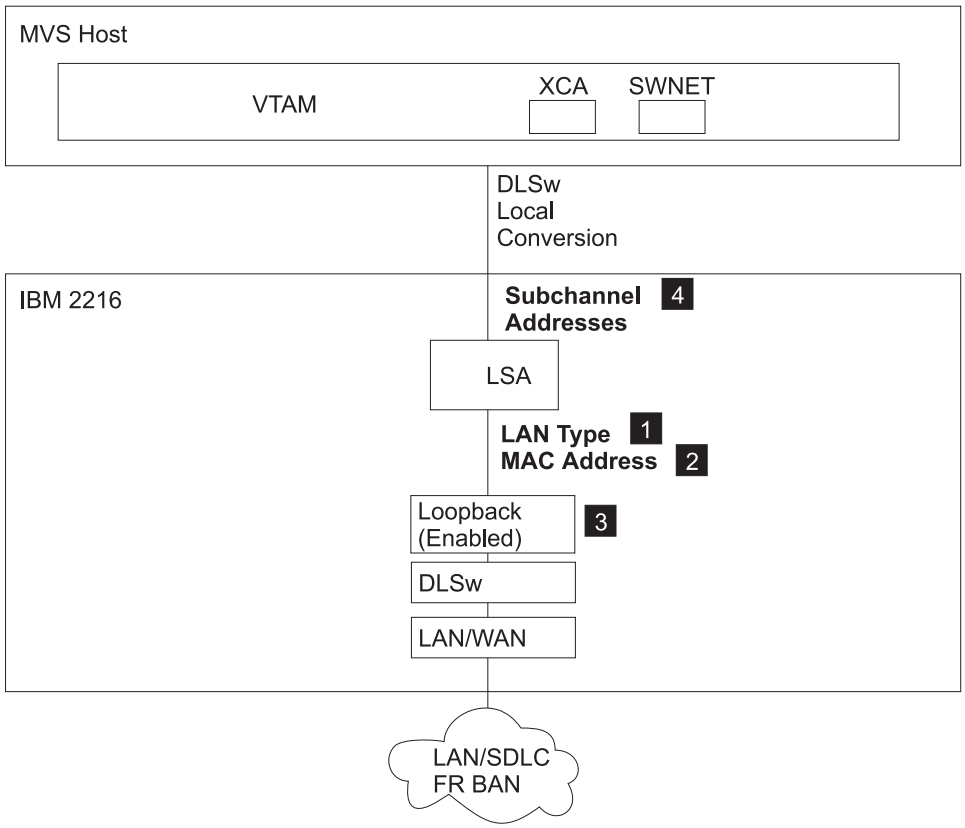



*Figure 31. Configuring Virtual Net Handlers for LSA DLSw Local Conversion*

Figure 31 shows a configuration that supports LSA DLSw local conversion. To configure a 2216 for LSA DLSw conversion:

1. Enable LSA loopback (3) using the **enable** command.
2. Configure the LAN type (1), either Ethernet or Token Ring. This is the frame type that the host expects to send and receive.
3. Configure a unique MAC address (2) to identify the host (VTAM) end of the loopback connection.

   **Note:** If the LAN type is Ethernet, then the MAC address must be in canonical format.
4. Configure the subchannel or subchannels (4) used by this connection as described in “Configuring an LSA Subchannel” on page 362.
5. Configure DLSw. Configuring DLSw involves enabling DLSw, setting the DLSw segment number, adding the local DLSw TCP partner and opening the service access points (SAPs) associated with the loopback interface that will be used for DLSw. Configure DLSw from the `config>` prompt.

   Enable DLSw, using the `enable dls` command.

   Set the DLSw segment number using the `set srb` command. The DLSw segment number must be unique. The DLSw segment number must be different from segment numbers assigned to other interfaces.

   Add the local DLSw TCP partner using the `add tcp` command.

Open the SAPs that will be used with the LSA loopback interface using the `open` command. The `open` command will prompt for an interface number. Input the interface number assigned to the LSA loopback interface that is defined for use with DLSW.

Opening SAPs and configuring bridging on the local LAN/WAN interfaces might be required. Opening the SAPs and configuring bridging permits incoming frames to be forwarded to DLSw.

For a description of the DLSw configuration parameters, refer to the chapter entitled ″Using and Configuring DLSw″ in the *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1*.

There are optional parameters:

**maxdata**
Maximum size of data handled by this virtual network.

**acklen**
The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

## Multi-Path Channel+ (MPC+) Support

Multi-Path Channel+ (MPC+) permits the VTAM host to communicate with the channel adapter in the 2216. An MPC+ Group is a set of subchannels, containing at least one read and one write subchannel, whose end points converge at a common MPC+ image in the VTAM Host.

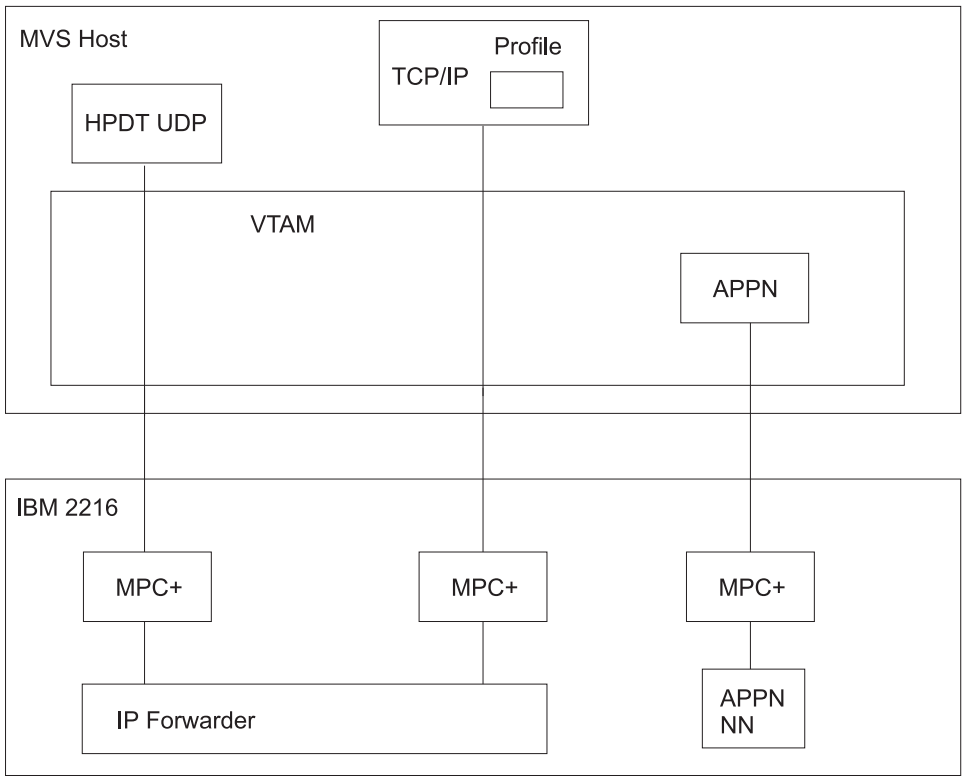



*Figure 32. Different types of MPC+ Connections.*

**Note:** UDP+ is not supported on a Parallel Channel Adapter (PCA).

Figure 32 shows the three types of MPC+ connections. The following sections describe the types in more detail:

- "Configuring the 2216 for APPN over MPC+" on page 348
- "Configuring the 2216 for UDP+ Over MPC+ (ESCON Channel Adapter Only)" on page 349
- "Configuring the 2216 for TCP/IP Over MPC+" on page 350

For information on the corresponding host definitions, see "Defining the 2216 to Host Programs" on page 322.

To configure an MPC+ Group in the 2216, configure an MPC+ interface on a base ESCON or PCA interface.

- An MPC+ connection to HPDT UDP in the host requires a dedicated MPC+ Group. An MPC+ Group that is dedicated cannot be shared by any other users or protocols.

  **Note:** UDP+ is not supported on a Parallel Channel Adapter (PCA).
- An MPC+ Group that is not dedicated can be shared by multiple TCP/IPs and by APPN.
- There can be multiple MPC+ Groups on a single channel adapter.

## Configuring the 2216 for APPN over MPC+

Figure 33 shows MPC+ flow and highlights key parameters at the host and in the 2216.

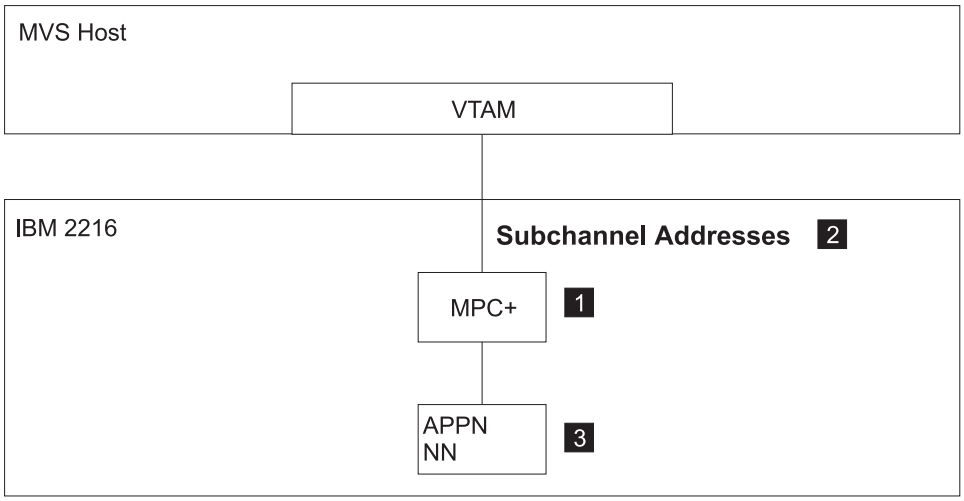



*Figure 33. Configuring Virtual Net Handlers for APPN over Multi-Path Channel+ (MPC+)*

Figure 33 shows the parameters required to configure MPC+ for APPN.

1 Configure the MPC+ Virtual Interface as described in “Configuring an MPC+ Virtual Interface” on page 364.

2 Configure subchannels for read and write connections to the host as described in “Configuring an MPC+ Subchannel” on page 365.

**Note:** Do not enable UDP+ exclusive use on the MPC+ interface. Ignore the disable outbound protocol data blocking parameter. It does not apply and has no effect for APPN.

There are optional parameters:

**reply timeout**
Timer for XID2/Disconnect timeout in milliseconds.

This is the amount of time that the MPC+ Group waits to hear from across the channel during XID2 and DISC exchanges before deciding that the other end of the channel is not answering and that this side should continue with the bring up or bring down of the MPC+ Group.

**sequencing interval timer**
Sequencing Interval Timer in milliseconds.

This timer is used to determine whether connection-oriented data is flowing smoothly across the connection on an MPC+ Group. The MPC+ control flows flow connection-oriented. Since these commands must have guaranteed delivery at the link level they flow connection-oriented and the Sequencing Interval timer is used to determine whether enough time has passed that checking of the delivery of connection-oriented traffic should be done.

**maxdata**
Maximum size of data handled by this virtual network handler.

**acklen**
The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

3 With the following exceptions, APPN is configured over the MPC+ interface as it is over other interface types:

- On the APPN "add port" command, specify link type MPC+.
- On the APPN "add port" command, you may specify the MPC+ sequencing interval timer.

For information on the corresponding host definitions, see "Configuring the VTAM Host for MPC+ for APPN" on page 333.

## Configuring the 2216 for UDP+ Over MPC+ (ESCON Channel Adapter Only)

Figure 34 graphically illustrates a UDP+ configuration over MPC+.

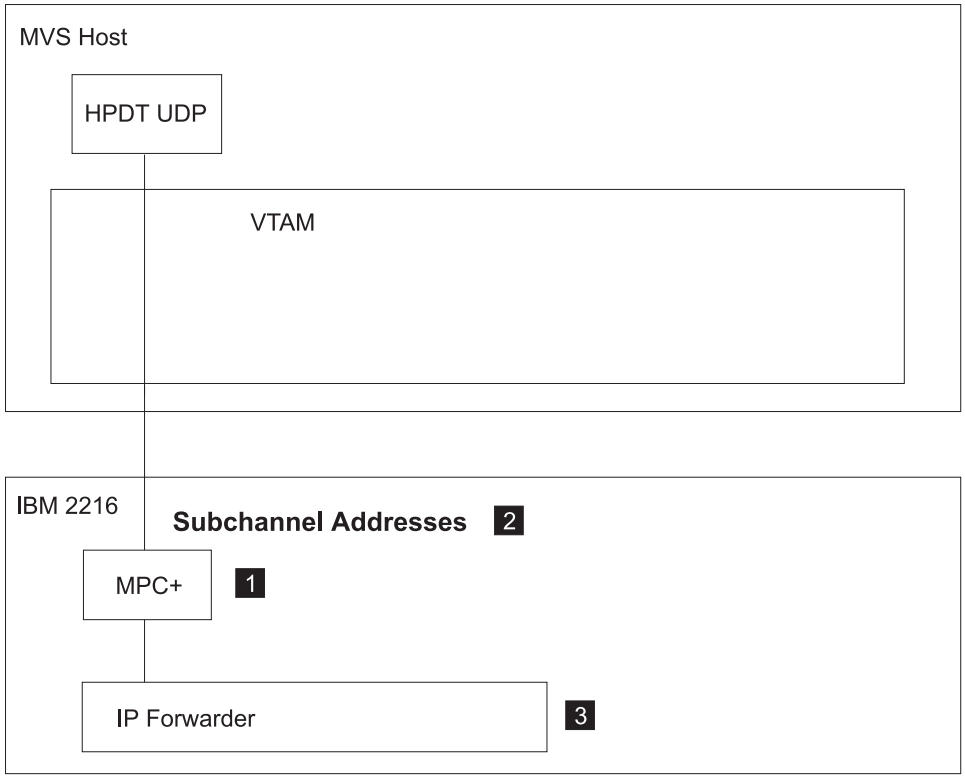



*Figure 34. Configuring Virtual Net Handlers for UDP+ over MPC+*

1 Configure the MPC+ Virtual Interface as described in "Configuring an MPC+ Virtual Interface" on page 364.

Enable UDP+ exclusive use on the MPC+ interface.

There are optional parameters:

**reply timeout**
Timer for XID2/Disconnect timeout in milliseconds.

This is the amount of time that the MPC+ Group waits to hear from across the channel during XID2 and DISC exchanges before deciding that the other end of the channel is not answering and that this side should continue with the bring up or bring down of the MPC+ Group.

**sequencing interval timer**
Sequencing Interval Timer in milliseconds.

This timer is used to determine whether connection-oriented data is flowing smoothly across the connection on an MPC+ Group. The MPC+ control flows are connection-oriented. Since these commands must have guaranteed delivery at the link level they flow connection-oriented and the Sequencing Interval timer is used to determine whether enough time has passed that checking of the delivery of connection-oriented traffic should be done.

**maxdata**
Maximum size of data handled by this virtual network handler.

**Note:** This value must equal the MTU coded for HPDT UDP in the host.

**acklen**
The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

**outbound protocol blocking**
Generally this parameter should be left enabled. See “Configuring an MPC+ Virtual Interface” on page 364 for details.

2 Configure subchannels for read and write connections to the host as described in “Configuring an MPC+ Subchannel” on page 365.

3 IP is configured over the MPC+ interface in the same manner that it is configured over other interface types; however:

- Only one IP address should be configured for a UDP+ MPC+ interface. This address must be equal to the destination_IP_address coded for HPDT UDP in the host.

  **Note:** If more than one IP address is configured, the last one configured is the one that is used.
- The source_IP_address coded for HPDT UDP in the host must be in the same IP subnetwork as the IP address configured in the 2216 on the UDP+ MPC+ interface.

For information on the corresponding host definitions, see “Configuring the Host for HPDT UDP:” on page 328.

## Configuring the 2216 for TCP/IP Over MPC+

Figure 35 on page 351 graphically illustrates a TCP/IP configuration over MPC+.

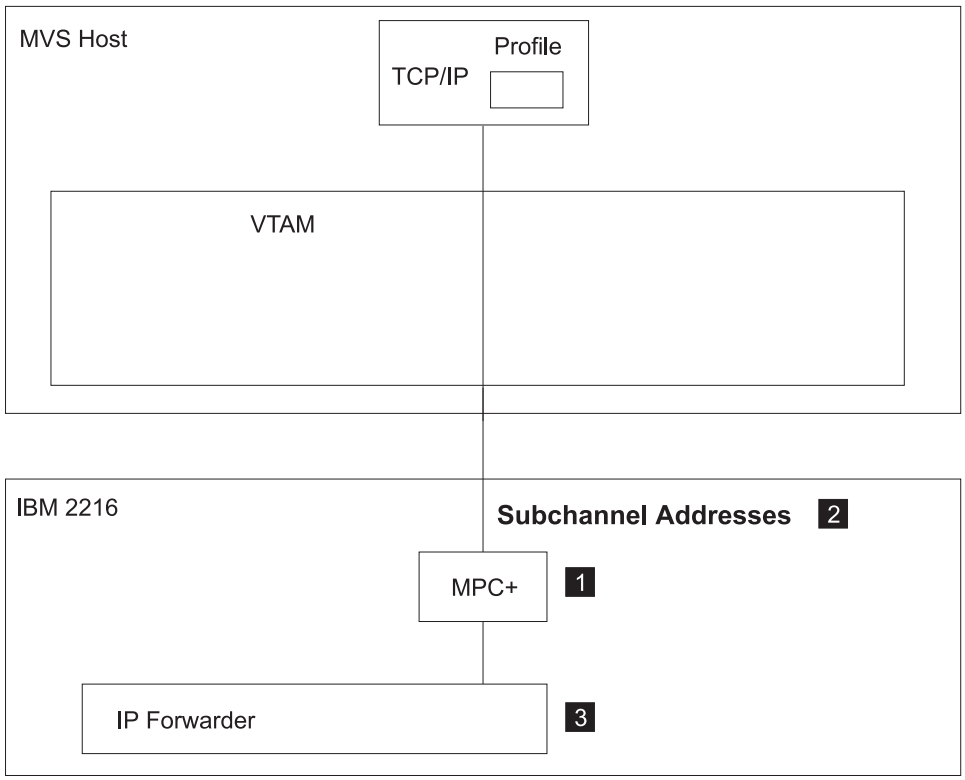



*Figure 35. Configuring Virtual Net Handlers for TCP/IP over MPC+*

1 Configure the MPC+ Virtual Interface as described in “Configuring an MPC+ Virtual Interface” on page 364.

**Note:** Do not enable UDP+ exclusive use on the MPC+ interface. TCP/IP is configured on an MPC+ interface by virtue of configuring IP addresses for the MPC+ network handler and having the MPC+ interface not be configured for the exclusive use of UDP+.

There are optional parameters:

**reply timeout**
Timer for XID2/Disconnect timeout in milliseconds.

This is the amount of time that the MPC+ Group waits to hear from across the channel during XID2 and DISC exchanges before deciding that the other end of the channel is not answering and that this side should continue with the bring up or bring down of the MPC+ Group.

**sequencing interval timer**
Sequencing Interval Timer in milliseconds.

This timer is used to determine whether connection-oriented data is flowing smoothly across the connection on an MPC+ Group. The MPC+ control flows flow connection-oriented. Since these commands must have guaranteed delivery at the link level they flow connection-oriented and the Sequencing Interval timer is used to determine whether enough time has passed that checking of the delivery of connection-oriented traffic should be done.

**maxdata**
Maximum size of data handled by this virtual network handler.

**Notes:**

1. This value must be less than or equal to the maximum amount of data the host can handle receiving over the channel (that is, MAXBFRU*4K, where MAXBFRU is from the VTAM TRLE corresponding to this MPC+ interface in the 2216).
2. The 2216 will never send an IP packet longer than maxdata across the MPC+ Group. However, depending upon the values for maxdata configured for other virtual network handlers using the same base channel interface as this MPC+ interface, the 2216 may actually accept a larger IP packet from the host.

**acklen**
The size (in bytes) of acknowledgment frames over this interface.

**blktimer**
The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

**outbound protocol blocking**
Generally this parameter should be left enabled. See “Configuring an MPC+ Virtual Interface” on page 364 for details.

2 Configure subchannels for read and write connections to the host as described in “Configuring an MPC+ Subchannel” on page 365.

3 IP is configured over the MPC+ interface in the same manner that it is configured over other interface types; however:

- Multiple TCP/IP over MPC+ interface connections can be established over a single MPC+ interface.
- In order to establish a TCP/IP over MPC+ connection, the TCP/IP HOME IP address must be in the same IP subnetwork as one of the IP addresses configured in the 2216 on the MPC+ interface.
- If multiple TCP/IP instances in the host are in the same IP subnetwork as configured for the 2216 MPC+ interface, multiple TCP/IP over MPC+ connections will be set up using the same 2216 IP address.

For information on the corresponding host definitions, see “Configuring the Host for TCP/IP” on page 322.

## Configuring the Channel Adapter Interface

The following steps are required to configure the ESCON or PCA interface:

1. Access the ESCON or PCA interface as described in “Accessing the Channel Interface” on page 355. This will cause the base interface to be defined.
2. Configure the virtual net handlers as described in:
   “Configuring an LCS Virtual Interface” on page 357
   “Configuring an LSA Virtual Interface” on page 360
   “Configuring an MPC+ Virtual Interface” on page 364
3. Configure the subchannels:
   “Configuring an LCS Subchannel” on page 358
   “Configuring an LSA Subchannel” on page 362

"Configuring an MPC+ Subchannel" on page 365

Once the 2216 ESCON or PCA configuration is complete,

- Configure the protocols.
- Save the configuration.
- Reboot the 2216 to activate changes.

## Planning for 2216 Support

# Chapter 33. Configuring and Monitoring the ESCON and Parallel Channel Adapters

This chapter describes the Enterprise Systems Connection (ESCON) and Parallel Channel Adapter configuration and operational commands. It includes the following sections:



For additional monitoring information, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*.

**Note About Examples**

Throughout this following sections there are examples of ESCON and PCA interface configurations. In areas where there is a significant difference between ESCON and PCA, there will be multiple examples.

## Accessing the Channel Interface

To access the channel adapter interface:

1. At the OPCON prompt, enter **talk 6**. 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, press **Return** again.
2. Enter the **list devices** command to display the network interface numbers that are currently configured.
3. Record the interface numbers.
4. Create a channel interface by either:
   - Entering the **add device esc** command at the `Config>` prompt to create an ESCON channel.

     ```
     Config> add dev esc
     Device Slot x(1-8) 1?
     Adding ESCON Adapter device in slot 1 port 1 as interface x
     ```

     **Note:** **x** is the assigned interface number.

     **Note:** The 2216 has eight slots, numbered 1 to 8.
   - Entering the **add device pca** command at the `Config>` prompt.

     ```
     Config> add dev pca
     Device Slot x(1-8) 1?
     Adding PCA device in slot 1 port 1 as interface x
     ```

**Note:** **x** is the assigned interface number.

**Note:** The 2216 has eight slots, numbered 1 to 8.

5. Enter the **network** command and the number of the interface you obtained in step 4. For example, if interface 0 is an ESCON interface:

```
Config> network 0
ESCON Config>
```

   The ESCON configuration prompt (`ESCON Config>`), is displayed.

6. Configure the channel adapter virtual net handlers and associated subchannels using the commands in Table 45.

## Channel Adapter Configuration Commands

The following commands can be entered at a channel adapter configuration prompt (either `ESC Config>` or `PCA Config>`):

*Table 45. Channel Interface Configuration Commands*

| Command | Description |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| add | Adds a virtual net handler for one of the base protocols or adds APPN loopback :<br>• LCS - LAN Channel Station Support<br>• LSA - Link Services Architecture<br>• MPC+ - Multi-Path Channel+<br>• APPN loopback<br>Each protocol provides a unique set of parameters which can be used to configure the virtual net handlers. |
| delete | Deletes an interface on the channel adapter. |
| list | Lists the channel adapter configuration and optionally lists subchannels. You can also list the transfer mode and channel transfer speed for a PCA. |
| mod | Changes the configuration of an interface on the channel adapter. |
| set | Sets the transfer mode and channel transfer speed for a parallel channel adapter (PCA).<br>**Note:** You can only access this command when configuring a PCA interface. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Add

Use the **add** command to add virtual network handlers for LCS, LSA, and MPC+, and to enable loopback for APPN.

**Syntax:**

**add** lcs
lsa

mpc

appn loopback

## Configuring an LCS Virtual Interface

Use the **add lcs** command to add an LCS virtual interface and get to the `ESCON Add Virtual>` or `PCA Add Virtual>`prompt from which you can enter other interface and subchannel parameters.

**Note:** Although LCS requires two subchannels, it is only necessary to specify one subchannel. An adjacent subchannel will be chosen such that the two subchannels will form a sequential pair with the write subchannel (device address is even) before the read subchannel (device address is odd).

After entering the **add lcs**, you will receive either the `ESCON Add Virtual>` or `PCA Add Virtual>` prompt. At this prompt you can enter the following commands:

**Command** **Description**

**acklen** *bytes* The size (in bytes) of acknowledgment frames over this interface.

**Note:** The default value of 10 causes acknowledgment frames to be blocked. This value gives the best performance in heavy traffic networks if it is used with the default blktimer value. For IP interactive traffic and bulk data transfer, set ***acklen*** to 100.

**Valid values:** 1 to 500 bytes

**Default value:** 10

**blktimer** *milliseconds*

The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

**Note:** The default value is set to give the best performance in heavy traffic networks. Set ***blktimer*** to 10 for IP interactive traffic and bulk data transfer.

**Valid values:** 1 to 20

**Default value:** 5

**lantype** *type* LAN type, either Ethernet, Token Ring, or FDDI

**Example:** Specifying LAN Type for an LCS interface on an ESCON interface

```
ESCON Add Virtual>lan
Please select one of the following LAN types:
    E  Ethernet
    T  Token Ring
    F  FDDI
LCS LAN Type: [E]?
```

**mac** *address* MAC Address of the virtual net handler

**maxdata** *bytes*

Maximum size of data handled by this virtual network handler.

**Valid Values:** 516 to 17749 for Token Ring, 1500 for Ethernet, 4478 for FDDI

**Default:** 2052 for Token Ring, 1500 for Ethernet, 4478 for FDDI

**subchannels** *command*
Places you at the next prompt based on the value of *command*. *Command* can be one of the following:

- add
- list
- mod

See "Configuring an LCS Subchannel" for the commands you can enter at these prompts and their description.

## Configuring an LCS Subchannel

Entering **subchannels** *command* places you at the next prompt based on the value of *command*. *Command* can be one of the following:

- add
- list
- mod
- exit

| Command | Description |
|---|---|
| **add** | Adds a subchannel pair and displays the `ESCON Config LCS Subchannel>` or `PCA Config LCS Subchannel>` prompt from which you can add Device Address, LPAR number, Link Address, and CU Logical Address. |

**Notes:**

1. You only need to specify the device address for PCA interfaces.
2. You must add or configure one subchannel for an LCS virtual interface. Although LCS requires two subchannels, it is only necessary to specify one subchannel. An adjacent subchannel will be chosen such that the two subchannels will form a sequential pair with the write subchannel (device address is even) before the read subchannel (device address is odd).

**device** *address*
The unit address transmitted on the channel path to select a 2216 device. It is also referred to as subchannel number in S/370 I/O architecture. It is a two-digit hexadecimal value that can range from X'00' to X'FF'. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.

**Valid Values:** X'00' to X'FF'

**Default:** None

**Important**
The following parameters do not apply for PCA.

**lpar** *number*
Logical partition number. This allows multiple logical host partitions (LPARs) to share one ESCON channel.

This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction.

If the host is not using EMIF, use the default of 0 for the LPAR number.

**Valid Values:** X'0' - X'F'

**Default:** X'0'

**link** *address*

If one ESCON Director (ESCD) is in the communication path, the link address is the ESCD port number that is attached to the host.

If two ESCDs are in the path, it is the host-side port number of the ESCD defined with the dynamic connection.

When no ESCD is in the communication path, this value must be set to X'01'.

**Valid Values:** X'01' - X'FE'

**Default:** X'01'

**cu** *address*

The Control Unit address defined in the host for the 2216. This value is defined in the host Input/Output Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction.

The Control Unit Address must be unique for each logical partition defined on the same host.

**Valid Values:** X'0' - X'F'

**Default:** X'0'

**Example:** Adding a subchannel for an LCS interface for an ESCON interface

```
ESCON Add Virtual>sub add
Please add or configure one subchannel for an LCS virtual interface.
Although LCS requires two subchannels, it is only necessary to specify
one subchannel.  An adjacent subchannel will be chosen such that the two
subchannels will form a sequential pair with the write subchannel (device
address is even) before the read subchannel (device address is odd).
ESCON Config LCS Subchannel>d 4
ESCON Config LCS Subchannel>e
```

**list** Lists information for the LCS subchannels.

**Example:** Listing subchannels for an ESCON LCS interface

```
ESCON Add Virtual>sub lis
         Read Subchannels:
         Sub  0   Device address   :  5   LPAR number        : 0
                  Link address     :  1   CU Logical Address : 0
         Write Subchannels:
         Sub  1   Device address   :  4   LPAR number        : 0
                  Link address     :  1   CU Logical Address : 0
```

**Example:** Listing subchannels for an PCA LCS interface

```
PCA Add Virtual>sub lis
          Read Subchannels:
          Sub  2   Device address   :  3

          Write Subchannels:
          Sub  3   Device address   :  2
```

**mod** Modifies a configured LCS subchannel pair. It lists the configuration for the configured LCS subchannels and allows you to modify one

of them by specifying the "sub" number from the list. Once you have selected the subchannel, you can change the device address, LPAR number, Link Address, and CU Logical Address as described in "Configuring an LCS Subchannel" on page 358 .

**Notes:**

1. You can only change the device address for PCA.
2. For ESCON or PCA, if there is only one subchannel configured, you can only modify the subchannel, not delete it.

## Configuring an LSA Virtual Interface

Use the **add lsa** command to add an LSA virtual interface and get to the `ESCON Add Virtual>` or `PCA Add Virtual>`prompt from which you can enter the following commands:

**Command** **Description**

**[enable or disable]**
Enables or disables loopback on an LSA interface.

**Note:** Only ***one*** of these parameters can be entered, depending on the state of the loopback function. If the loopback is disabled, you can enable it; if it is enabled, you can disable it.

**Valid Values:** enable or disable

**Default:** disable

**acklen** *bytes* The size (in bytes) of acknowledgment frames over this interface.

**Note:** The default value of 10 causes acknowledgment frames to be blocked. This value gives the best performance in heavy traffic networks if it is used with the default blktimer value. Set ***acklen*** to 100 for IP bulk data transfer and interactive traffic.

**Valid values:** 1 to 500 bytes

**Default value:** 10

**blktimer** *milliseconds*
The maximum time (in milliseconds) to wait before sending an unfull data block to the host.

**Note:** The default value is set to give the best performance in heavy traffic networks. Set ***blktimer*** to 10 for IP bulk data transfer and interactive traffic over Ethernet and token-ring LANs.

**Valid values:** 1 to 20

**Default value:** 10

**lantype** *type* LAN type, either Ethernet, or Token-Ring.

**mac** *address* A unique MAC address to identify this virtual interface. This

parameter is available only when loopback is enabled. It is the MAC address of the LSA/VTAM side of the loopback connection. The MAC address of the APPN side of the loopback connection is specified using ADD APPN.

**maxdata** *bytes*

Maximum size of data handled by this virtual network handler.

**Valid Values:** 516 to 17749 for Token Ring, 1500 for Ethernet

**Default:** 2052 for Token Ring, 1500 for Ethernet

**net** *interface#* This parameter is available only when loopback is disabled. It is used to indicate the LAN adapter over which this LSA net will communicate. The LAN adapter must have been previously configured and can only be Token-Ring, Ethernet (including emulated LANs), or FDDI.

**subchannels** Places you at the next prompt based on the value of *command*. *Command* can be one of the following:

- add
- delete
- list
- mod

See “Configuring an LSA Subchannel” on page 362 for the commands you can enter at these prompts and their description.

There are four types of LSA connections, as shown in Figure 27 on page 340. They are:

- “Configuring an LSA Direct Connection at the 2216” on page 341
- “Configuring an LSA APPN Connection at the 2216” on page 342
- “Configuring an LSA DLSw Connection at the 2216” on page 343
- “Configuring an LSA DLSw Local Conversion at the 2216” on page 345

The example shows adding two LSA interfaces. The first one uses loopback and the second one is a direct connection.

**Example 1:** Adding an ESCON LSA interface with loopback

```
ESCON Config>add lsa
ESCON Add Virtual>enable
Enabling loopback through network 2.
Please set the MAC address using the "MAC" command
ESCON Add Virtual>mac 40:00:00:00:22:16
ESCON Add Virtual>lan
Please select one of the following LAN types:
    E  Ethernet
    T  Token Ring
LSA LAN Type: [E]? e
ESCON Add Virtual>sub add
ESCON Add LSA Subchannel>link c5
ESCON Add LSA Subchannel>d 8
ESCON Add LSA Subchannel>e
ESCON Add Virtual>e
ESCON Config>list all
Net:  2   Protocol: LSA    LAN type: LSA Ethernet    LAN number:  0
          Maxdata: 1500
          Loopback is enabled.
          MAC address: 400000002216
          Block timer:    10 ms  ACK length:  10 bytes
          Sub  0   Dev addr:  8  LPAR: 0  Link addr: C5  CU addr: 0
```

**Example 2:** Adding a PCA LSA interface with direct connection

```
PCA Config>add lsa
PCA Add Virtual>net 0
PCA Add Virtual>sub add
PCA Add LSA Subchannel>d 7
PCA Add LSA Subchannel>e
PCA Add Virtual>e
PCA Config>list all
Net:  6   Protocol: LSA    LAN type: Token Ring      LAN number:  0
          Maxdata: 2052
          Loopback is not enabled.
          MAC address: Obtained from net 0
          Block timer:    10 ms  ACK length:  10 bytes
          Sub  0   Dev addr:  7
PCA Config>
```

## Configuring an LSA Subchannel

Entering **subchannels** *command* places you at the next prompt based on the value of *command*. *Command* can be one of the following:

- add
- delete
- list
- mod
- exit

| Command | Description |
|---|---|
| **add** | Adds a subchannel and displays the `ESCON Add LSA Subchannel>` or `PCA Add LSA Subchannel>` prompt from which you can add: |

**device** *address*

The unit address transmitted on the channel path to select a 2216 device. It is also referred to as subchannel number in S/370 I/O architecture. It is a two-digit hexadecimal value that may range from 00-FF. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.

**Valid Values:** X'00' to X'FF'

**Default:** None

**Important**

The following parameters do not apply for PCA.

**lpar** *number*

Logical partition number. This allows multiple logical host partitions, (LPARs) to share one ESCON channel.

This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction.

If the host is not using EMIF, use the default of 0 for the LPAR number.

**Valid Values:** X'0' - X'F'

**Default:** X'0'

**link** *address*

If one ESCON Director (ESCD) is in the communication path, the link address is the ESCD port number that is attached to the host.

If two ESCDs are in the path, it is the host-side port number of the ESCD defined with the dynamic connection.

When no ESCD is in the communication path, this value must be set to X'01'.

**Valid Values:** X'01' - X'FE'

**Default:** X'01'

**cu** *address*

The Control Unit address defined in the host for the 2216. This value is defined in the host Input/Output Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction. The Control Unit Address must be unique for each LPAR defined on the same host.

**Valid Values:** X'0' - X'F'

**Default:** X'0'

Enter **exit** to return to the previous prompt.

**Example:** Adding a subchannel for an ESCON LSA interface

```
ESCON Add Virtual>sub add
ESCON Add LSA Subchannel>link f7
ESCON Add LSA Subchannel>device 0
ESCON Add LSA Subchannel>cu 0
ESCON Add LSA Subchannel>lpar 0
ESCON Add LSA Subchannel>exit
```

**Example:** Adding a subchannel for a PCA LSA interface

```
PCA Add Virtual>sub add
PCA Add LSA Subchannel>device 2
PCA Add LSA Subchannel>exit
```

**delete** Deletes a configured LSA subchannel. It lists the configuration for the configured LSA subchannels and allows you to delete one of them by specifying the “sub” number from the list.

**list** Lists information for the LSA subchannels.

**Example:** Listing subchannels for an ESCON LSA interface

```
ESCON Config Virtual>sub list
         Sub  0   Device address   : 42   LPAR number        : 0
                  Link address     : C5   CU Logical Address : 0
         Sub  1   Device address   : 43   LPAR number        : 0
                  Link address     : C5   CU Logical Address : 0
         Sub  2   Device address   : 44   LPAR number        : 0
                  Link address     : C5   CU Logical Address : 0
```

**Example:** Listing subchannels for a PCA LSA interface

```
PCA Config Virtual>sub list
          Sub  0   Device address   : B
          Sub  1   Device address   : 12
          Sub  2   Device address   : 10
          Sub  3   Device address   : A
          Sub  4   Device address   : C
          Sub  5   Device address   : E
```

**mod** Modifies a configured LSA subchannel. It lists the configuration for the configured LSA subchannels and allows you to modify one of them by specifying the “sub” number from the list. Once you have

selected the subchannel, you can change the device address, LPAR number, Link Address, and CU Logical Address as described in "Configuring an LSA Subchannel" on page 362.

**Notes:**

1. You can only change the device address for PCA.
2. For ESCON or PCA, if there is only one subchannel configured, you can only modify the subchannel, not delete it.

## Configuring an MPC+ Virtual Interface

Use the **add mpc** command to add an MPC+ virtual interface and get to the `ESCON Add Virtual>` or `PCA Add Virtual>` prompt from which you can enter other interface and subchannel parameters:

| Command | Description |
|---|---|
| **acklen** *bytes* | The size (in bytes) of acknowledgment frames over this interface.<br><br>**Note:** The default value of 10 causes acknowledgment frames to be blocked. This value gives the best performance in heavy traffic networks if it is used with the default blktimer value.<br><br>**Valid values:** 1 to 500 bytes<br><br>**Default value:** 10 |
| **blktimer** *milliseconds* | The maximum time (in milliseconds) to wait before sending an unfull data block to the host.<br><br>**Note:** The default value is set to give the best performance in heavy traffic networks.<br><br>**Valid values:** 1 to 20<br><br>**Default value:** 5 |
| **disable_outbound protocol blocking** | Prevents the MPC+ Group from blocking multiple protocol packets within a single MPC+ packet when sending data to the host.<br><br>**Note:** This parameter affects UDP+ and TCP/IP traffic. Enabling this parameter gives better performance in heavy traffic networks. Enable is the default. |
| **disable udp+ exclusive use** | To undedicate the MPC+ interface to UDP+.<br><br>**Notes:**<br>1. This parameter applies to ESCON interfaces only.<br>2. This means that only HPDT UDP in the host will be able to establish a connection across this MPC+ Group. UDP+ can never share an MPC+ Group with other protocols (for example, APPN, TCP/IP). |

**enable_outbound protocol blocking**
Enables the MPC+ Group to block multiple protocol packets within a single MPC+ packet when sending data to the host.

**Note:** This parameter affects UDP+ and TCP/IP traffic. Enabling this parameter gives better performance in heavy traffic networks. Enable is the default.

**enable udp+ exclusive use**
To dedicate the MPC+ interface to UDP+.

**Notes:**

1. This parameter applies to ESCON interfaces only.
2. This means that only HPDT UDP in the host will be able to establish a connection across this MPC+ Group. UDP+ can never share an MPC+ Group with other protocols (for example, APPN, TCP/IP).

**maxdata** *bytes*
Maximum size of data handled by this virtual network handler.

**Valid Values:** 512 to 32 768

**Default:** 2 048

**reply timeout** *milliseconds*
Timer for XID2/Disconnect timeout in milliseconds.

**Valid Values:** 1 to 50 000

**Default:** 45 000

**sequencing interval timer** *milliseconds*
Sequencing Interval Timer in milliseconds.

**Valid Values:** 1 to 50 000

**Default:** 3000

**subchannels** Places you at the next prompt based on the value of *command*. *Command* can be one of the following:

- addr
- addw
- delete
- list
- mod
- exit

See “Configuring an MPC+ Subchannel” for the commands you can enter and their description.

## Configuring an MPC+ Subchannel

**Note:** A subchannel defined as a read subchannel to VTAM is a write subchannel to the 2216 and a subchannel defined as a write subchannel to VTAM is a read subchannel to the 2216.

Entering **subchannels** *command* places you at the next prompt based on the value of *command*. *Command* can be one of the following:

- addr

## Channel Adapter Configuration Commands (Talk 6)

- addw
- delete
- list
- mod
- exit

| Commands | Description |
|---|---|
| **addr** | Adds a read subchannel and displays the `ESCON Add MPC+ Read Subchannel>` or `PCA Add MPC+ Read Subchannel>` prompt from which you can enter the following commands: |

| Command | Description |
|---|---|
| **device** *address* | The unit address transmitted on the channel path to select a 2216 device. It is also referred to as subchannel number in S/370 I/O architecture. It is a two-digit hexadecimal value that may range from 00-FF. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.<br><br>**Valid Values:** X'00' to X'FF'<br><br>**Default:** None |

**Important**
The following parameters do not apply for PCA.

| | |
|---|---|
| **lpar** *number* | Logical partition number. This allows multiple logical host partitions, (LPARs) to share one ESCON channel.<br><br>This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction.<br><br>If the host is not using EMIF, use the default of 0 for the logical partition number.<br><br>**Valid Values:** X'0' - X'F'<br><br>**Default:** X'0' |
| **link** *address* | If one ESCON Directors (ESCD) is in the communication path, the link address is the ESCD port number that is attached to the host.<br><br>If two ESCDs are in the path, it is the host-side port number of the ESCD defined with the dynamic connection.<br><br>When no ESCD is in the communication path, this value must be set to X'01'.<br><br>**Valid Values:** X'01' - X'FE'<br><br>**Default:** X'01' |
| **cu** *address* | The Control Unit address defined in the host for the 2216. This value is defined in the host Input/Output |

Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction.

The Control Unit Address must be unique for each LPAR defined on the same host.

**Valid Values:** X'0' - X'F'

**Default:** X'0'

Enter **exit** to return to the `ESCON Add Virtual>` prompt.

**Example:** Adding read subchannels for a PCA MPC+ interface

```
PCA Add Virtual>sub addr
PCA Add MPC+ Read Subchannel>d 8
PCA Add MPC+ Read Subchannel>e
PCA Add Virtual>sub addr
PCA Add MPC+ Read Subchannel>d 9
PCA Add MPC+ Read Subchannel>e
```

**addw** Adds a write subchannel and displays the `ESCON Add MPC+ Write Subchannel>` or `PCA Add MPC+ Write Subchannel>` prompt from which you can enter the following commands:

| Command | Description |
|---|---|
| **device** *address* | The unit address transmitted on the channel path to select a 2216 device. It is also referred to as subchannel number in S/370 I/O architecture. It is a two-digit hexadecimal value that may range from 00-FF. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.<br>**Valid Values:** X'00' to X'FF'<br>**Default:** None |

**Important**

The following parameters do not apply for PCA.

| | |
|---|---|
| **lpar** *number* | Logical partition number. This allows multiple logical host partitions, (LPARs) to share one ESCON channel.<br>This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction.<br>If the host is not using EMIF, use the default of 0 for the LPAR number.<br>**Valid Values:** X'0' - X'F'<br>**Default:** X'0' |
| **link** *address* | If one ESCON Director (ESCD) is in the communication path, the link address is the ESCD port number that is attached to the host.<br>If two ESCDs are in the path, it is the host-side port number of the ESCD defined with the dynamic connection. |

When no ESCD is in the communication path, this value must be set to X'01'.

**Valid Values:** X'01' - X'FE'

**Default:** X'01'

**cu** *address* The Control Unit address defined in the host for the 2216. This value is defined in the host Input/Output Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction.

The Control Unit Address must be unique for each logical partition defined on the same host.

**Valid Values:** X'0' - X'F'

**Default:** X'0'

**Example:** Adding write subchannels for an ESCON MPC+ interface

```
ESCON Add Virtual>sub addw
ESCON Add MPC+ Write Subchannel>d 10
ESCON Add MPC+ Write Subchannel>e
ESCON Add Virtual>sub addw
ESCON Add MPC+ Write Subchannel>d 11
ESCON Add MPC+ Write Subchannel>e
```

**delete** Deletes a configured MPC+ subchannel. It lists the configuration for the configured MPC+ subchannels and allows you to delete one of them by specifying the “sub” number from the list.

**list** Lists information for the MPC+ subchannels.

**Example:** Listing subchannels for an ESCON MPC+ interface

```
ESCON Add Virtual>sub lis
         Read Subchannels:
         Sub  0   Device address   :  8   LPAR number        : 0
                  Link address     :  1   CU Logical Address : 0
         Sub  1   Device address   :  9   LPAR number        : 0
                  Link address     :  1   CU Logical Address : 0
         Write Subchannels:
         Sub  2   Device address   : 10   LPAR number        : 0
                  Link address     :  1   CU Logical Address : 0
         Sub  3   Device address   : 11   LPAR number        : 0
                  Link address     :  1   CU Logical Address : 0
```

**Example:** Listing subchannels for a PCA MPC+ interface

```
PCA Add Virtual>sub lis
          Read Subchannels:
          Sub  0   Device address   : 12
          Sub  1   Device address   : 13

          Write Subchannels:
          Sub  2   Device address   : 14
          Sub  3   Device address   : 15
```

**mod** Modifies a configured MPC+ subchannel. It lists the configuration for the configured MPC+ subchannels and allows you to modify one of them by specifying the “sub” number from the list. Once you have selected the subchannel, you can change the device address, LPAR number, Link Address, and CU Logical Address as described in “Configuring an MPC+ Subchannel” on page 365 .

**Note:** You can only change the device address for PCA.

Once you have returned to the previous prompt, you can list the entire MPC+ configuration as shown in the following example:

**Example:** Listing and changing an ESCON MPC+ configuration

```
ESCON Config>list all
Net:  1   Protocol: MPC+   LAN type: MPC+            LAN number:  0
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          MPC Group is for exclusive use of UDP+
          Outbound protocol data blocking is enabled
          Block timer:   5 ms    ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 40  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 41  LPAR: 0  Link addr: F5  CU addr: 0

Net:  2   Protocol: MPC+   LAN type: MPC+            LAN number:  1
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          Outbound protocol data blocking is enabled
          Block timer:   5 ms    ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 42  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 43  LPAR: 0  Link addr: F5  CU addr: 0

Net:  3   Protocol: MPC+   LAN type: MPC+            LAN number:  2
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          Outbound protocol data blocking is enabled
          Block timer:   5 ms    ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 44  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 45  LPAR: 0  Link addr: F5  CU addr: 0

ESCON Config>mod 3
ESCON Config Virtual>?
REply timeout
SEQuencing int timer
MAXdata
SUBchannels
Exit
ESCON Config Virtual>rep 3100
ESCON Config Virtual>exit
ESCON Config>list all
Net:  1   Protocol: MPC+   LAN type: MPC+            LAN number:  0
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          MPC Group is for exclusive use of UDP+
          Outbound protocol data blocking is enabled
          Block timer:   5 ms    ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 40  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 41  LPAR: 0  Link addr: F5  CU addr: 0

Net:  2   Protocol: MPC+   LAN type: MPC+            LAN number:  1
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          Outbound protocol data blocking is enabled
          Block timer:   5 ms    ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 42  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 43  LPAR: 0  Link addr: F5  CU addr: 0

Net:  3   Protocol: MPC+   LAN type: MPC+            LAN number:  2
          Maxdata: 2048
          Reply TO: 3100   Sequencing Interval Timer: 3000
          Outbound protocol data blocking is enabled
          Block timer:   5 ms    ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 44  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 45  LPAR: 0  Link addr: F5  CU addr: 0

ESCON Config>
```

## Configuring APPN Loopback

**Important:** You only need to configure APPN loopback once for each physical 2216. If you configure APPN loopback on an ESCON channel, you do not need to enable it on a PCA channel in the same 2216.

**Notes:**

1. APPN loopback cannot be added unless loopback has been enabled on an LSA virtual net as described in "Configuring an LSA Virtual Interface" on page 360.
2. You can only add APPN loopback once on a given IBM 2216.

Use the **add appn** command to add APPN loopback. You will get either the `ESCON Add Virtual>` or `PCA Add Virtual>` prompt where you can enter the following commands:

| Commands | Description |
|---|---|
| **lantype** *type* | Ethernet or Token Ring |
| **mac** *address* | A unique MAC address to identify the APPN side of the loopback connection in the 2216. This address must be different from the MAC address given to the host (VTAM) side of the loopback connection when configuring the LSA interface. |

**Example:** Adding APPN loopback to an ESCON interface

```
ESCON Config>add appn
ESCON Add Virtual>
ESCON Add Virtual>lan
Please select one of the following LAN types:
    E  Ethernet
    T  Token Ring
APPN LAN Type: [T]?
ESCON Add Virtual>mac
MAC address in 00:00:00:00:00:00 form [000000000000]?
     40:00:22:16:00:09
ESCON Add Virtual>e
ESCON Config>li all
Net:  9   Protocol: APPN Loopback   LAN type: Token-Ring/802.5
          APPN loopback MAC address: 400022160009

Net:  5   Protocol: LSA    LAN type: Token Ring           LAN number:  0
          Maxdata: 2052
          Loopback is enabled.
          MAC address: 400022160005
          Block timer:    10 ms  ACK length:  10 bytes
          Sub  0   Dev addr:  0  LPAR: 0  Link addr:  1  CU num: 0

Net:  6   Protocol: LSA    LAN type: Token Ring           LAN number:  1
          Maxdata: 2052
          Loopback is not enabled.
          MAC address: Obtained from net 3
          Block timer:    10 ms  ACK length:  10 bytes
          Sub  0   Dev addr:  1  LPAR: 0  Link addr:  1  CU num: 0

Net:  7   Protocol: LCS    LAN type: LCS Ethernet 802.3   LAN number:  0
          Maxdata: 1500
          MAC address: 400022160007
          Block timer:     5 ms  ACK length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr:  5  LPAR: 0  Link addr:  1  CU num: 0
          Write Subchannels:
          Sub  1   Dev addr:  4  LPAR: 0  Link addr:  1  CU num: 0
ESCON Config>e
```

**Notes:**

1. The APPN port would be configured on net 9 of the example.

2. Configure APPN link stations that will connect to VTAM over the ESCON channel to use the MAC address of the LSA net as the destination MAC address. Do not use the APPN loopback net for this purpose.
3. Any LSA nets connection to APPN must have the same LAN type as the APPN loopback net.

## Delete

Use the **delete** command to delete an interface on the channel adapter. If you know the interface number you wish to delete, you can specify it; otherwise, if you do not enter an interface number, the configuration is listed and you will be prompted to enter an interface number.

**Syntax:**

**delete** *interface_number*

(no parameter)

**interface_number**
Deletes the configuration for the specified interface number.

**(no parameters)**
Lists the configured interfaces for the channel adapter and prompts you for the interface number you wish to delete.

**Example:** Deleting an interface (no parameters given)

```
PCA Config>del

Net:  5   Protocol: APPN Loopback   LAN type: Token-Ring/802.5
          APPN loopback MAC address: 400000000406
Net:  2   Protocol: LSA    LAN type: Token Ring      LAN number:  0
          Maxdata: 2052
          Loopback is enabled.
          MAC address: 400000000403
          Block Timer:   10 ms   ACK length:   10 bytes
Net:  3   Protocol: LSA    LAN type: Token Ring      LAN number:  1
          Maxdata: 2052
          Loopback is not enabled.
          MAC address: Obtained from net 1
          Block Timer:   10 ms   ACK length:   10 bytes
Net:  4   Protocol: MPC+   LAN type: MPC+            LAN number:  0
          Maxdata: 2048
          Reply TO: 3100   Sequencing Interval Timer: 3000

          Outbound protocol data blocking is enabled
          Block Timer:    5 ms   ACK length:   10 bytes
Virtual net number to delete: [2]? 3
Are you sure?(Yes or [No]): y
```

## Mod

Use the **mod** command to modify a configured interface on the channel adapter. If you know the interface number you wish to modify, you can specify it; otherwise, if you do not enter an interface number, the configuration is listed and you will be prompted to enter an interface number.

**Syntax:**

**modify** *interface_number*

(no parameters)

**interface_number**
Modifies the configuration for the specified interface number.

**(no parameters)**

Lists the configured interfaces for the channel adapter and prompts you for the interface number you wish to modify.

**Example:**

```
ESCON Config> mod
Net:  1   Protocol: MPC+   LAN type: MPC+            LAN number:  0
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          MPC Group is for exclusive use of UDP+
          Outbound protocol data blocking is enabled
          Block timer:     5 ms  ACK length:  10 bytes
Net:  2   Protocol: MPC+   LAN type: MPC+            LAN number:  1
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          Outbound protocol data blocking is enabled
          Block timer:     5 ms  ACK length:  10 bytes
Virtual net number to configure: [1]? 2
ESCON Config Virtual> ?
REply timeout
SEQuencing int timer
MAXdata
SUBchannels
Exit
ESCON Config Virtual>re
Reply Time Out (range 1-50000 milliseconds): [45000]? 30003
ESCON Config Virtual>sub list
          Read Subchannels:
          Sub  0   Device address   :  7   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Write Subchannels:
          Sub  1   Device address   :  6   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
ESCON Config Virtual>sub addr
ESCON Add MPC+ Read Subchannel> ?
LINk address (ESCD Port)
LPAR number
CU logical address
Device address
Exit
ESCON Add MPC+ Read Subchannel>d 5
ESCON Add MPC+ Read Subchannel>? e
ESCON Config Virtual>sub list
          Read Subchannels:
          Sub  0   Device address   :  7   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Sub  1   Device address   :  5   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Write Subchannels:
          Sub  2   Device address   :  6   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
ESCON Config Virtual>sub ?
ADDRead subchannel
ADDWrite subchannel
MODify subchannel
DELete subchannel
LIst subchannels
ESCON Config Virtual>sub del
          Read Subchannels:
          Sub  0   Device address   :  7   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Sub  1   Device address   :  5   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Write Subchannels:
          Sub  2   Device address   :  6   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
Subchannel number to delete: [0]? 0
Are you sure?(Yes or [No]): y
ESCON Config Virtual>sub list
          Read Subchannels:
          Sub  0   Device address   :  5   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Write Subchannels:
          Sub  1   Device address   :  6   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
```

```
ESCON Config Virtual>sub mod
          Read Subchannels:
          Sub  0   Device address   :  5   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Write Subchannels:
          Sub  1   Device address   :  6   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
Subchannel number to modify: [0]? 1
ESCON Modify MPC+ Subchannel>d 2
ESCON Modify MPC+ Subchannel>e
ESCON Config Virtual>sub list
          Read Subchannels:
          Sub  0   Device address   :  5   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
          Write Subchannels:
          Sub  1   Device address   :  2   LPAR number        : 0
                   Link address     : F4   CU Logical Address : 0
ESCON Config Virtual> exit
ESCON Config>
```

## List (ESCON)

Use the **list** command to list the channel adapter configuration and also (with **list all**) list a subchannel summary.

**Syntax:**

| | |
|---|---|
| **list** | (no parameters) |
| | all |

**(no parameters)**

Lists the channel adapter configuration.

**Example:** Listing an ESCON configuration

```
ESCON Config>li
Net:  5   Protocol: LSA    LAN type: Token Ring          LAN number:  0
          Maxdata: 2052
          Loopback is enabled.
          MAC address: 400022160005
          Block timer:    10 ms  ACK length:  10 bytes
```

**all** Lists the channel configuration with a subchannel summary. Two examples are provided. The first is for a channel adapter with LSA and LCS subchannels. The second is for a channel adapter with MPC+ subchannels.

**Example for LSA and LCS:** Listing an ESCON configuration with subchannel summary

```
ESCON Config>li all
Net:  2   Protocol: LCS    LAN type: LCS FDDI    LAN number:  0
          Maxdata: 4478
          MAC address: 400000002216
          Block Timer:  5 ms  ACK Length:  10 bytes
          Sub  0   Dev addr:  8  LPAR: 0  Link addr: C5  CU addr: 0

Net:  5   Protocol: LSA    LAN type: Token Ring          LAN number:  0
          Maxdata: 2052
          Loopback is enabled.
          MAC address: 400022160005
          Block timer:    10 ms  ACK length:  10 bytes
          Sub  0   Dev addr:  0  LPAR: 0  Link addr:  1  CU addr: 0
```

**Example for MPC+:** Listing an ESCON configuration with subchannel summary

```
Net:  1   Protocol: MPC+   LAN type: MPC+           LAN number:  0
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          MPC Group is for exclusive use of UDP+
          Outbound protocol data blocking is enabled
          Block Timer:  5 ms  ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 40  LPAR: 0  Link addr: F5  CU addr: 0
```

```
          Write Subchannels:
          Sub  1   Dev addr: 41  LPAR: 0  Link addr: F5  CU addr: 0

Net:  2   Protocol: MPC+   LAN type: MPC+           LAN number:  1
          Maxdata: 2048
          Reply TO: 45000   Sequencing Interval Timer: 3000
          Outbound protocol data blocking is enabled
          Block Timer:  5 ms  ACK Length:  10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 42  LPAR: 0  Link addr: F5  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 43  LPAR: 0  Link addr: F5  CU addr: 0
```

## List (PCA)

Use the **list** command to list the transfer mode and channel transfer speed configured or the virtual interfaces configured.

**Syntax:**

**list** base

virtual

**base** Lists the transfer mode and channel transfer speed configured.

**virtual [all or (no parameter)]**
Lists a summary of the configuration for the virtual interfaces or the configuration of all virtual interfaces and their subchannels (**all**).

## Set (PCA Only)

Use the **set** command to set the transfer mode and channel transfer speed for a parallel channel adapter (PCA).

**Syntax:**

**set** tmode *value*

**tmode** *value*
Specifies the mode of transfer that the 2216 uses to transfer data to the host, either DC interlock or Data Streaming and the channel transfer speed when using data streaming.

**Valid values:**

**D** Specifies Direct-coupled (DC) Interlock mode. This mode is the standard I/O interface that requires a response to a demand.

**S** Specifies speeds less than or equal to a 3.0 MB data streaming mode.

**S4** Specifies speeds less than or equal to a 4.5 MB data streaming mode.

**Default value:** D

# Accessing the Channel Interface Monitoring Process

To access the ESCON or PCA interface:

1. At the OPCON prompt, enter **talk 5**. For example:

```
* talk 5
+
```

2. To display the monitoring prompt for the channel interface or any of the channel adapter's virtual interfaces, enter the **network** command followed by the interface number of the interface.

   If you do not know the interface number, use the **configuration** command at the + prompt to display a list of interface numbers configured on the router.

```
Multiprotocol Access Services

5765-D47 Feature 2804 V3.1 Mod 0 PTF 0 RPQ 0 MAS.BF1 cc_115a test-load


Num Name  Protocol
0   IP     DOD-IP
3   ARP    Address Resolution
11  SNMP   Simple Network Management Protocol
23  ASRT   Adaptive Source Routing Transparent Enhanced Bridge
28  APPN   Advanced Peer-to-Peer Networking [HPR]
29  NHRP   Next Hop Routing Protocol
30  APPN   Advanced Peer-to-Peer Networking [ISR]

Num Name  Feature
2   MCF    MAC Filtering
7   CMPRS  Data Compression Subsystem
8   NDR    Network Dispatching Router
10  AUTH   Authentication

31 Networks:
Net Interface  MAC/Data-Link         Hardware                      State
 0   TKR/0      Token-Ring/802.5      Token-Ring                    Up
 1   Eth/0      Ethernet/IEEE 802.3   Ethernet                      Up
 2   PCA/0      Parallel Channel      Parallel Channel              Up
 3   LCS/0      LCS                   Parallel Channel              Up
 4   MPC/0      MPC                   Parallel Channel              Up
 5   LSA/0      LSA                   Parallel Channel              Up
 6   TKR/1      Token-Ring/802.5      APPN Loopback                 Up
 7   ESCON/0    ESCON                 ESCON Channel                 Up
 8   MPC/1      MPC                   ESCON Channel                 Up
 9   LCS/1      LCS                   ESCON Channel                 Up
 10  LSA/1      LSA                   ESCON Channel                 Up
```

## Channel Interface Monitoring Commands

The following commands can be entered at the channel adapter monitoring prompt (ESCON> or PCA>):

*Table 46. Channel Interface Monitoring Commands*

| Command | Description |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| List | Lists subchannels or lists nets. |
| Net | Lists a specific network interface. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## List

Use the **list** command to list all subchannels or to list all network interfaces. You can also list the speed of a PCA interface.

**Syntax:**

| | |
|---|---|
| **list** | base (PCA only) |
| | nets |
| | subchannels |

**base** Lists the transfer mode and channel transfer speed of the channel adapter.

**nets** Lists network interfaces.

**Example:** List networks

```
PCA>li ne
+ net 1
PCA Base Monitoring
PCA> li nets
Net: 2   Type: LSA   LAN Type: Token-Ring/802.5  LAN Number: 0
          Net state: Up
Net: 4   Type: LSA   LAN Type: Token-Ring/802.5  LAN Number: 1
          Net state: Up
Net: 5   Type: LCS   LAN Type: FDDI              LAN Number: 0
          Net state: Down
```

**Type** Type of virtual interface: LCS, LSA, or MPC+

**LAN Type**
LAN type, either Token-Ring/802.5, Ethernet/802.3, Ethernet/V2, Ethernet or FDDI.

**Note:** This field is not displayed for subchannels that are part of an MPC+ interface.

**LAN Number**
Interface number of the LAN

**Note:** This field is not displayed for subchannels that are part of an MPC+ interface.

**Group Number**
The group number is used internally by the device to identify a virtual MPC+ net interface on the channel adapter.

**Net State**
State of the network: Up, Down, Disabled, Not Present, HW Mismatch, or Testing.

**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 network interface's adapter is not present.

**HW Mismatch**
An adapter other than a channel adapter is located in the slot or the physical channel adapter installed is not the same type as the configured channel adapter.

**Note:** Only base nets will have the states "Not present" and "HW mismatch."

**Testing**
The system is attempting to determine if a network connection exists

**subchannels**
Lists subchannels

```
ESCON> li sub
The following  subchannels are defined:
       Local address: 00    Device address: 00   CU Logical Address: 00
                            Link: C5    LPAR: 00
                            Type: LSA
```

```
            The following lantypes/lannums are using this subchannel:
                              LAN type: Token-Ring/802.5 LAN number: 0
Local address: 01    Device address: DD   CU Logical Address: 0B
                     Link: 5C    LPAR: 02
                     Type: LSA
            The following lantypes/lannums are using this subchannel:
                              LAN type: Token-Ring/802.5 LAN number: 0
Local address: 02    Device address: 07   CU Logical Address: 00
                     Link: C5    LPAR: 00
                     Type: LSA
            The following lantypes/lannums are using this subchannel:
                              LAN type: Token-Ring/802.5 LAN number: 1
Local address: 03    Device address: 02   CU Logical Address: 00
                     Link: C5    LPAR: 00
                     Type: LCS
            The following lantypes/lannums are using this subchannel:
                              LAN type: FDDI             LAN number: 0
Local address: 04    Device address: 03   CU Logical Address: 00
                     Link: C5    LPAR: 00
                     Type: LCS
            The following lantypes/lannums are using this subchannel:
                              LAN type: FDDI             LAN number: 0
```

**Local Address**
The subchannel address index used internally by the device.

**Device Address**
The unit address transmitted on the channel path to select a device. It is also referred to as subchannel number in S/370 I/O architecture. It is a two-digit hexadecimal value that may range from X'00' to X'FF'. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.

**Note:** CU Logical Address, Link Address, and LPAR are only displayed for ESCON adapters.

**CU Logical Address**
The Control Unit address defined in the host for the device. This value is defined in the host Input/Output Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction.

The Control Unit Address must be unique for each LPAR defined on the same host.

**Link Address**
If one ESCON Director (ESCD) is in the communication path, the link address is the ESCD port number that is attached to the host.

If two ESCDs are in the path, the link address is the host-side port number of the ESCD defined with the dynamic connection.

When no ESCD is in the communication path, this value must be set to 0x01.

**LPAR** Logical partition number. This allows multiple partitions in a logically partitioned (LPAR) host to share one ESCON fiber.

This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction.

If the host is not using EMIF, the LPAR number is 0 (zero).

**Type** Type of virtual interface: LCS, LSA, or MPC+

**LAN Type**
LAN type, either Token-Ring/802.5, Ethernet/802.3, Ethernet/V2, Ethernet or FDDI.

**Note:** This field is not displayed for subchannels that are part of an MPC+ interface.

**LAN Number**
Interface number of the LAN

**Note:** This field is not displayed for subchannels that are part of an MPC+ interface.

**Group Number**
The group number is used internally by the device to identify a virtual MPC+ net interface on the channel adapter.

## Net

Use the **net** command to get to the monitoring environment for one of the virtual interfaces as described in:

- "Channel Adapter LCS Interface Monitoring Commands"
- "Channel Adapter LSA Interface Monitoring Commands" on page 380
- "Channel Adapter MPC+ Interface Monitoring Commands" on page 381

**Syntax:**

**net** *net_number*

# Channel Adapter LCS Interface Monitoring Commands

The following commands can be entered at the LCS monitoring prompt (LCS>):

*Table 47. Channel Adapter LCS Interface Monitoring Commands*

| Command | Description |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| List | Lists subchannels or lists nets. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## List

Use the **list** command to display information for an LCS interface.

**Syntax:**

**list**

**Example:**

```
LCS> list
LCS Virtual Adapter
LCS Information for Net 4
--- ----------- --- --- --
LAN Type: Token-Ring/802.5     LAN Number: 0
Local Read Subchannel number:  7
```

```
Local Write Subchannel number: 6
MAC Address: 400022160001
Local IP Address: 9.192.200.1
Status: Down
```

**LAN Type**
LAN type, either Token-Ring/802.5, Ethernet/802.3, Ethernet/V2, Ethernet or FDDI.

**Note:** This field is not displayed for subchannels that are part of an MPC+ interface.

**LAN Number**
Interface number of the LAN

**Note:** This field is not displayed for subchannels that are part of an MPC+ interface.

**Read Subchannel**
The local subchannel from which the device receives data.

**Write Subchannel**
The local subchannel through which the device transmits data.

**MAC Address**
A unique MAC address to identify this virtual interface.

**Local IP address**
IP Address that was assigned to this network interface.

**Status**
Status of the network: Up, Down, Disabled, Not Present, HW Mismatch, or Testing

**Up** The network connection is established.

**Down** The network connection can not be determined.

**Disabled**
Device is disabled and diagnostic testing can be performed

**Not Present**
Indicates that the network interface's adapter is not present.

**HW Mismatch**
An adapter other than a channel adapter is located in the slot or the physical channel adapter installed is not the same type as the configured channel adapter.

**Note:** Only base nets will have the states "Not present" and "HW mismatch."

**Testing**
The system is attempting to determine if a network connection exists

# Channel Adapter LSA Interface Monitoring Commands

The following commands can be entered at the LSA monitoring prompt (LSA>):

*Table 48. Channel Adapter LSA Interface Monitoring Commands*

| Command | Description |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists information for adapters, SAPs, or link stations. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to display information for adapters, SAPs, and link stations.

**Syntax:**

**list** adapter
sap
link stations

**adapter**
Lists virtual adapters for LSA.

**Example:** List Virtual Adapter for LSA

```
LSA> list ad
LSA Virtual Adapter
LSA Information for Net 2
--- ----------- --- --- --
LAN Type: Token-Ring/802.5     LAN Number: 0
MAC Address: 4000000000CF
Downstream network: Loopback - Net 2
Status: Host connected

#SAPs Open: 1     #Link Stations Open: 1
Maximum frame size: 2052 (0x804)
Host User ID                  Subchannel
------------                  ----------
  00000000                       0

1 host user(s)
```

**#SAPs Open**
Number of SAPs opened by VTAM on this LSA interface

**#Link Stations Open**
Number of link stations open for all SAPs on this LSA interface

**Maximum Frame Size**
Maximum frame size supported over this LSA interface

**Host User ID**
A unique ID generated by VTAM to identify the host user on a given subchannel

**Subchannel**
The local subchannel being used by this host user.

**sap** Lists Service Access Points (SAPS) for LSA

**Example:** List SAP for LSA

```
LSA> list sap
 SAP    Provider     User     Max Link    Open Link
Number   SAP ID     SAP ID    Stations    Stations
------  --------   --------   --------    ---------
  4    02000000   00000001      1           1
1 SAPs currently open
```

**SAP Number**
Identifies the SAP to LLC

**Provider SAP ID**
A unique ID generated by VTAM to identify this SAP

**User SAP ID**
A unique ID generated by device to identify this SAP

**Max Link Stations**
Maximum number of link stations VTAM can open on this SAP

**Open Link Stations**
Number of link stations currently open on this SAP

**link** Lists link information for LSA

**Example:** List link for LSA

```
LSA> list link
Please specify a SAP number (0-236): [4]? 4
                Link Stations on SAP 4

Station    Destination    Destination      Link     Frames      Frames
   ID      MAC Address    SAP Number      Status     Sent      Received
-------    -----------    -----------     ------    ------     --------

02000001   40000000ABCD        4         Connected        9            9
1 link station(s) open on SAP 4
```

**Station ID**
A unique ID generated by device to identify this link station

**Destination MAC Address**
MAC address of the remote LLC link station

**Destination SAP Number**
SAP value of the remote LLC link station

**Link Status**
Current status of the LLC connection

**Frames Sent**
Number of packets sent to VTAM for this link station

**Frames Received**
Number of packets received from VTAM for this link station

## Channel Adapter MPC+ Interface Monitoring Commands

The following commands can be entered at the MPC+ monitoring prompt (MPC+>):

*Table 49. Channel MPC+ Interface Monitoring Commands*

| Command | Description |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists subchannels. |

*Table 49. Channel MPC+ Interface Monitoring Commands (continued)*

| Command | Description |
|---|---|
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to display the MPC+ Group, subchannels, Connection Manager (CM), and connection information.

**Syntax: list** cm
connection
mpc group
subchannel

**cm** Shows information on the Connection Manager that is running on the MPC+ Group. The information shown is the group token, the type of connection manager, and the current state.

The states are:

**Reset** The CM is currently inactive.

**Pending Active-waiting for MPC+ Group**
The underlying MPC+ Group is coming active.

**Pending Active-waiting for this side**
The other side has initiated bring-up of the CM but this side has not started to initiate bring-up of the CM.

**Pending Active-waiting for other side**
This side has initiated bring-up of the CM but the other side has not started to initiate bring-up of the CM.

**Pending Active-callee**
This side is waiting for the other side, which is the caller in this bring-up to start the call.

**Pending Active-caller**
This side has called out to the other side and is waiting for the other side to respond to the call.

**Active** Active and usable

**Example:** List an active CM for MPC+

```
MPC+>li cm
MPC+ Connection Managers(CM)
              Group Token    type      state
------------------------ ------- ----------

090144953400000009            PTP Active
```

**Example:** List, no active CM for MPC+

```
MPC+>li cm
No CMs on this MPC+ Group
```

**connection**
Shows information on the connections running on the MPC+ Group/Connection Manager. The information shown is in two parts: the virtual circuit and the connections under the virtual circuit. The following

information is shown for the virtual circuit: the local and remote virtual circuit tokens, the protocol type, and the current state.

**Local Virtual Circuit Token**
Token in the device representing this virtual circuit.

**Remote Virtual Circuit Token**
Token in the host representing this virtual circuit. This field is blank if not known.

**Protocol**
The upper layer protocol that this virtual circuit is using.

**States for the Virtual Circuit**
The states for the virtual circuit are:

**Reset** The virtual circuit is currently inactive.

**Active-other side**
The other side is currently accepting calls (connections) for this virtual circuit.

**Active-this side**
This side is currently accepting calls (connections) for this virtual circuit.

**Active-both sides**
Both sides of the virtual circuit are accepting calls (connections) for this virtual circuit.

**Not accepting new calls**
This connection is not accepting new calls (connections). However, connections that are already running on the virtual circuit will stay up.

The information shown for the connection is the local and remote connection tokens and the current state. For IP protocols (that is, UDP+ and TCP/IP), the local and destination IP addresses associated with the connection will also be shown, if known.

**Local Connection Token**
The token in the device representing this connection.

**Remote Connection Token**
The token in the host representing this connection. This field is blank if not known.

**States for the Connection**
The states for the connection are:

**Reset** The connection is currently inactive

**Pending Active - callee**
This side is about to respond to the call request from the other side.

**Pending Active - caller**
This side has called out to the other side and is waiting for the other side to response to the call.

**Pending Active - awaiting datastart**
The connection is waiting for both sides to be ready to start allowing user data to flow.

**Active** Active and usable

**Local IP address**
The IP address on the MPC+ interface in the device that is associated with this connection. This field will be displayed only for IP protocols.

**Destination IP Address**
The IP address in the host that is associated with this connection. This field will be displayed only for IP protocols.

**Example:** List, active connections for MPC+

```
MPC+>li conn
MPC+ Connections
Virtual Circuit Token = 090144C22C0000000D
Remote Registration Token(s) = 05000101A5
Protocol =       APPN, State = Active-both sides
                       Local Connection Token = 090144C3300000000E
                       Remote Connection Token = 05000101A6
                       State =     Active

Protocol =     TCP/IP, State = Active-both sides
                       Local Connection Token = 090144C4400000000F
                       Remote Connection Token = 05000101B0
                       State =     Active
                       Local IP address = 100.0.0.1
                       Destination IP address = 100.0.0.2
```

**Example:** List, no connections active for MPC+

```
MPC+>li conn
No User Connections on this MPC+ Group
```

**mpc** Displays information about the MPC+ Group. It displays the local and remote registration token, if known, and the current state of the MPC+ Group. Also, if the MPC+ is for the exclusive use of UDP+, that will be indicated. If the MPC+ Group is not for the exclusive use of UDP+, exclusive use will not be mentioned in the display.

**Note:** UDP+ is not supported on a Parallel Channel Adapter (PCA).

**Example:**

```
MPC+>li mpc
MPC+ Group
Local registration token = 0901422A3C00000000
Remote registration token = 050001019D
state = Active
This MPC+ Group is for the exclusive use of UDP+.
Outbound protocol data blocking is enabled for the MPC+ Group.
```

**Local registration token**
Token in the device representing this MPC+ Group.

**Remote registration token**
Token in the host representing this MPC+ Group. This field is blank if not known.

**State** The state of the MPC+ Group:

**Reset** The MPC+ Group is currently inactive.

**Pending Active-xid2(00)**
In the process of becoming active and currently processing xid2(00)s.

**Pending Active-xid2(07)**
In the process of becoming active and currently processing xid2(07)s.

**Active** Active and usable.

**Pending Reset**
Pending inactive (in other words, in the process of coming down).

**subchannel**

Shows information about the subchannels that are part of the MPC+ Group. It shows the local subchannel number, logical partition number, Link address, Control Unit (CU) logical address, Device Address, type of Subchannel (READ or WRITE), and the current state of the subchannel. The type should be the opposite of what is configured at the host.

**Example:** List subchannels for MPC+

```
MPC+>li sub
MPC+ Subchannels
 Local          Link  CU Log.  Device
 number  LPAR  addr  address  address   type      state
 ------- ----- ----- ------   ------- ---------- ----------

      1     0    F4      0       9       READ     Active
      0     0    F4      0       8      WRITE     Active
```

**Note:** Only device address is displayed when listing a PCA subchannel.

**Local number**
The subchannel address index used internally by the device.

**Note:** CU Logical Address, Link Address, and LPAR are only displayed for ESCON adapters.

**LPAR** Logical partition number. This allows multiple partitions in a logically partitioned (LPAR) host to share one ESCON fiber.

This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction.

If the host is not using EMIF, the LPAR number is 0 (zero).

**Link Address**
If one ESCON Director (ESCD) is in the communication path, the link address is the ESCD port number that is attached to the host.

If two ESCDs are in the path, the link address is the host-side port number of the ESCD defined with the dynamic connection.

When no ESCD is in the communication path, this value must be set to 0x01.

**CU Logical Address**
The Control Unit address defined in the host for the device. This value is defined in the host Input/Output Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction.

The Control Unit Address must be unique for each logical partition defined on the same host.

**Device Address**
The unit address transmitted on the channel path to select a device. It is also referred to as subchannel number in S/370 I/O architecture. It is a two-digit hexadecimal value that may range from X'00' to X'FF'. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the

real device. It is a two-digit hexadecimal value that may range from X'00' to X'FF'. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.

**Type** Whether this is a read or write subchannel.

**state** The state of the subchannel:

**Reset** The subchannel is currently inactive.

**Pending Active-xid2(00)**
The subchannel is becoming active and currently processing xid2(00)s.

**Pending Active-xid2(07)**
The subchannel becoming active and currently processing xid2(07)s.

**Active** The subchannel is active and part of an MPC+ Group.

**Pending Reset**
The subchannel is pending inactive (in other words, in the process of coming down).

# Chapter 34. 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" on page 388

**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 18 for a table of protocols and the interfaces that support those protocols.

## Accessing the Interface Configuration Process

See "Accessing Network Interface Configuration and Operating Processes" on page 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 35. Using the X.25 Network Interface" on page 389 .

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.

## Clocking and Cable Type

This section applies to all uses of a serial port for: FR, PPP, X.25, SDLC Relay, and SDLC.

If a modem or CSU/DSU is attached to the serial port then the router is taking on the DTE role in terms of clocking on the line, so configure a DTE cable type and external clocking.

If you want to attach two routers directly without a modem, CSU/DSU, or modem eliminator, then one of the routers will take on the DCE role in terms of clocking on

the line. Connect a direct attach cable to the router that will act as the DCE and configure the following parameters for its serial interface.

1. A DCE cable type
2. Internal clocking
3. The clocking/line speed

The other router will take on the DTE role in terms of clocking and should be configured as if it were attached to a modem or CSU/DSU

**Note:** Configuring a DTE as opposed to a DCE cable has no impact on whether or not the WAN net handler takes on the peer device. For example, the router always acts as a Frame Relay DTE device and uses a FR UNI interface even when a Frame Relay interface is configured to use a DCE cable.

## 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 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands.

# Chapter 35. Using 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:



For information on configuring X.25 Transport Protocol (XTP) for transporting X.25 traffic over TCP/IP, see "Chapter 37. Using XTP" on page 431.

## 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** *#* 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. For DDN networks, use the **add htf-addr** and the **set htf-addr** commands to convert the protocol address associated with this interface to the X.121 address format required for DDN address translation. 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 27 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 **national-personality set** 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 50. 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<br>low 2-way=1, high 2-way=64<br>low outbound=0, high outbound=0 |

*Table 50. Set Command (continued)*

| Parameter | Default |
|---|---|
| throughput-class ... | inbound=outbound=2400 |
| vc-idle ... | 30 |

*Table 51. National Enable Parameters*

| Parameter | DDN Default | GTE Default |
|---|---|---|
| accept-reverse-charges | off | on |
| bi-cug | off | off |
| bi-cug-with-outgoing-access | off | off |
| cug | off | off |
| cug-deletion | off | off |
| cug-insertion | off | off |
| cug-with-incoming-access | off | off |
| cug-with-outgoing-access | off | off |
| cug-zero-override | off | off |
| flow-control-negotiation | on | on |
| frame-ext-seq-mode | off | off |
| packet-ext-seq-mode | off | off |
| request-reverse-charges | off | on |
| suppress-calling-addresses | off | off |
| throughput-class-negotiation | on | on |
| truncate-called-addresses | off | off |

*Table 52. 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 |
| 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 |

*Table 52. National Set Parameters (continued)*

| Paramter | DDN Default | GTE Default |
|---|---|---|
| collision-timer | 10 seconds | 10 seconds |
| standard-version | 1984 | 1984 |
| t1-timer | 4 seconds | 4 seconds |
| t2-timer | 0 | 0 |
| truncate-called-addr-size | 2 | 2 |

## Null Encapsulation

Null Encapsulation is to allows the user to multiplex multiple network layer protocols over one X.25 circuit. This function may be used to avoid using an unreasonable number of virtual circuits.

## Limitations

Null Encapsulation is not supported for QLLC. This function is supported for SVC (Switched Virtual Circuits).

## Configuration changes

The encapsulation option NULL has been added for the following T6 commands:

Under X25 config: add address IP (may input enc type = NULL)

Under X25 config: add address IPX (may input enc type = NULL)

Under X25 config: add address DNA (may input enc type = NULL)

Under X25 config: add address VINES (may input enc type = NULL)

Under X25 config: list addr will show active enc type = NULL if the priority 1 type is NULL.

T5 commands:

Under X25 ıint▪: List SVCS will include enc type = NULL

## Configuring Null Encapsulation and Closed User Groups (CUG)

Since More than one Protocol can run over one virtual circuit while using Null Encapsulation, the CUG(s) defined for each protocol over that circuit must be the same. It is strongly suggested that the user configure multiple Protocols same destination as follows:

Configure CUG using the add address. The CUG(s) defined must be the same for each protocol defined at the same address.

If the CUG is defined at the add protocol level, The CUG(s) must be the same for all peers. (This method is more restrictive).

Configure CUG at the interface level. This insures all peers have the same CUG values. (This method is the most restrictive)

Any of the above methods may be used as long as any incoming call CUG definition must be valid for all protocols sharing that circuit. Valid means that the CUG was defined for the specific address or was defaulted to use either the protocol or interface circuit definition.

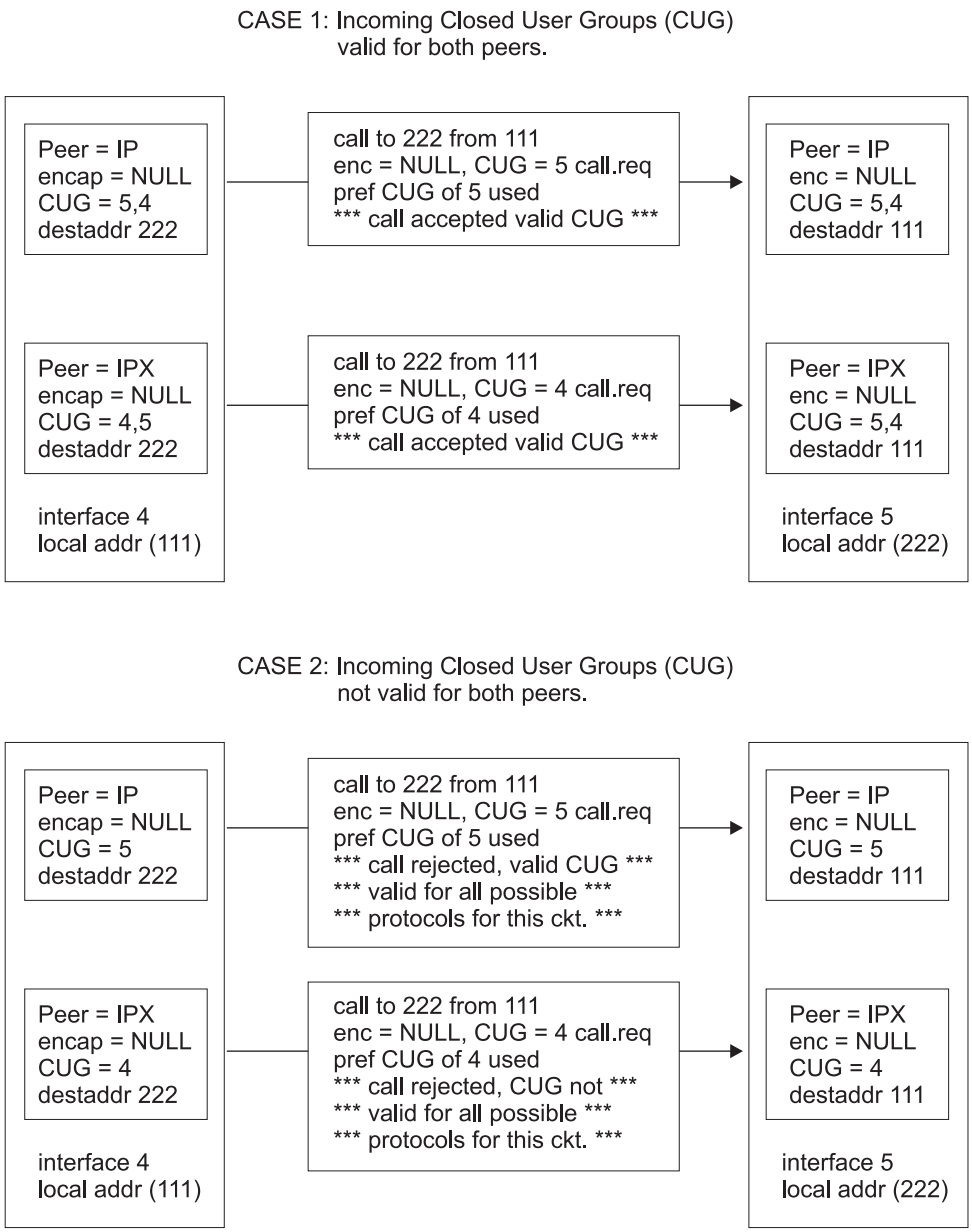



*Figure 36. Closed User Group Null Encapsulation*

## Understanding Closed User Groups

A *closed user group (CUG)* is a group of X.25 DTEs allowed to establish connections with other specific DTEs. CUG numbers are defined by your network provider and you can only use the CUGs the provider assigns you. You can configure an address-specific CUG, a protocol-specific CUG, or an interface-specific CUG. If all of three types of CUG numbers are configured for a DTE, the closed user group facility uses the address-specific destination CUG in a call request when contacting another DTE. If only a protocol-specific and an interface-specific CUG are configured for a DTE, the closed user group facility uses the protocol-specific CUG in a call request when contacting another DTE.

A single DTE can belong to multiple CUGs. You must specify a preferred CUG for that DTE. The preferred CUG is used when the router initiates calls to other DTEs. A single DTE cannot have more than a total of 5 preferred or normal closed user groups.

## Bilateral Closed User Groups

A *bilateral closed user group (BCUG)* is a closed user group consisting of only two DTEs. The DTEs within the BCUG can originate calls to members of the BCUG and any DTEs that are not members of any CUG or BCUG. A single DTE cannot have more than a total of 5 preferred or normal bilateral CUGs.

A DTE uses a BCUG to establish circuits in the same way the DTE uses CUGs to establish circuits (see Table 53), however, if both a BCUG and a CUG is defined for an interface, protocol, or address, the BCUG is used to establish the circuit.

## Types of Extended Closed User Groups

The following extensions to closed user groups are supported:

**CUG with Outgoing Access**
The DTE can belong to one or more CUGs. The DTE can originate calls to members of the CUG and to any DTE belonging to other CUGs with Incoming Access.

**CUG with Incoming Access**
The DTE can belong to one or more CUGs. The DTE can receive calls from DTEs not belonging to any CUG or from DTEs belonging to other CUGs with Outgoing Access.

**BCUG with Outgoing Access**
The DTE can belong to one or more BCUGs. The DTE can originate calls to members of the BCUG and to any DTE not belonging to any BCUG.

## Establishing X.25 Circuits with Closed User Groups on a Device

When you have enabled the closed user group facility, and a DTE receives a call request, it uses the CUG in the call request to determine whether to accept or reject the call from the DTE. If the CUG in the call request does not match a configured CUG on the interface, protocol, or on the destination associated with the calling DTE, the request is rejected. Table 53 summarizes how X.25 circuits are established based on CUGs, if the interface, protocol, and address CUG numbers are different and incoming access is not enabled.

*Table 53. Establishing Incoming X.25 Circuits for Closed User Groups*

| Incoming Call Request Contains | Receiving DTE CUG Definition | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Interface CUG Only | Protocol CUG Only | Address Specific CUG | Interface and Protocol CUG | Interface and Address CUG | Protocol and Address CUG | All CUGs | No CUGs |
| No CUG | Reject | Reject | Reject | Reject | Reject | Reject | Reject | Accept |
| Interface CUG | Accept | Reject | Reject | Reject | Reject | Reject | Reject | Reject |

*Table 53. Establishing Incoming X.25 Circuits for Closed User Groups (continued)*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Protocol CUG | Reject | Accept | Reject | Accept | Reject | Reject | Reject | Reject |
| Address Specific CUG | Reject | Reject | Accept | Reject | Accept | Accept | Accept | Reject |

For outgoing calls on an interface, if you have enabled either the CUG or the BCUG facility, each call request will contain the configured preferred CUG (if any) for the destination or, if no address-specific CUG is configured, the CUG used is the CUG defined for the protocol, or if no protocol-specific CUG is configured, the CUG used is the CUG defined for the interface. If no CUG number has been configured, the CUG facility is not included in any outgoing call request.

### Overriding Closed User Group Processing for CUG 0

You can configure the DTE such that it does not validate incoming calls with a CUG of 0 in the call request. This ability allows you to permit specific calls to complete even when you have not enabled incoming access. Using the **national enable cug 0 override** command forces the device to ignore the CUG facility if the CUG number is 0. The call request will not be compared with any configured CUG number.

## Configuring X.25 Closed User Groups

To use closed user groups on X.25 interfaces:

1. Request CUG numbers from your network provider. You will need these numbers when configuring X.25.
2. Enable the closed user group facility using the **national enable cug** command and related commands.
3. Enable the bilateral closed user group facility, if desired, using the **national enable bi-cug** command and related commands.
4. Configure the appropriate CUG numbers for the DTEs. Specify the preferred CUG, CUG, preferred bilateral CUG, and bilateral CUG, as needed. This is done through the **add address** command.
5. Configure the appropriate CUG and bilateral CUG for the protocol, if required. This is done through the **add protocol** command.

   **Note:** You should only configure these CUGs if you are restricting all X.25 circuits established over the X.25 interface for this protocol to DTEs belonging to this set of unique CUGs or BCUGs unless you override it with an address-specific CUG.
6. Configure the appropriate CUG and bilateral CUG for the interface, if required. This is done through the **add cug** command.

   **Note:** You should only configure these CUGs if you are restricting all X.25 circuits established over the X.25 interface to DTEs belonging to this set of unique CUGs or BCUGs unless uyou override it with an address or protocol-specific CUG.

# Chapter 36. Configuring and Monitoring the X.25 Network Interface

This chapter describes the X.25 configuration and operational commands and includes the following sections:



## 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 54 shows the commands.

*Table 54. X.25 Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Set | Sets the local and DDN X.25 node addresses, window size for 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. Sets speed, encoding, clocking, throughput class, and cable type. |
| 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 | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

# 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:**

**set**

- address . . .
- cable
- calls-out . . .
- clocking . . .
- default-window-size . . .
- encoding
- equipment-type . . .
- htf addr . . .
- inter-frame-delay . . .
- mtu
- national-personality . . .
- pvc . . .
- speed . . .
- svc
- throughput-class . . .
- vc-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.

**calls-out** *value*

Sets the maximum number of locally initiated, simultaneously active SVCs.

**Valid Values:** 1 to 239

**Default Value:** 4

**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 55 on page 401.

**Default:** external

**default-window-size** *value*

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

**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 (must be DTE for X.31)

**htf addr** *x.25-node-addr*

Sets the local DTE address when DDN is used. It converts the IP address to an X.121 address as opposed to the **set address** command, which is used to set the local DTE address when CCITT is used.

**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 IBM 2216 requests from 0 to 15 extra flags between frames.

**Note:** If you configure a non-zero inter-frame delay for a X.25 interface on the 8-port EIA-232E adapter, 6-port V.35/V.36 adapter, or 8-port X.21 adapter, you configure the speed using the **set speed** command.

**Default:** 0

**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 to 16384.

**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 acknowledgment.

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.

**Notes:**

1. The MTU parameter has significant impact on the memory requirements and memory utilization of the device. Use an MTU value of 8192 or less for devices with less than 8M of memory.
2. The amount of memory available while the device is running limits the number of SVCs that can be established and still maintain optimal performance. For recommendations on the maximum number of SVCs see the product home page on the World Wide Web.

**national-personality** *GTE-Telenet* **or** *DDN*

Sets the 28 default parameters for either GTE-Telenet or DDN National Personality.

**Default:** GTE-Telenet

**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 400 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.

**Valid values:** Use Table 55 to determine the clock speeds you can set for the various adapters.

*Table 55. 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.

*Table 56. 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 256 000 bps.

**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 the range of logical channel numbers to be assigned to inbound SVCs. By default, there are no inbound-only SVCs.

**Valid values:** 0 to 4095

**Default values:** 0

**Two-way**

Specifies the range of logical channel numbers to be assigned to two-way SVCs. By default, there are sixty-four 2-way SVCs.

**Valid values:** 0 to 4095

**Default values:**

**svc low**

1

**svc high**

64

**Outbound**
Specifies the range of logical channel numbers to be assigned to outbound SVCs. 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 57 shows a possible VC configuration.

*Table 57. Example VC Definitions*

| | 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 requests.

**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

# 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:**

**enable** ddn—address-translations

**Note:** Enabling ddn-address-translations is no longer allowed. This feature defaults to enabled when the national personality selected is DDN, and defaults to disabled in all other cases.

incoming-calls-barred

lower-dtr

outgoing-calls-barred

**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.

**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.

**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.

## 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.

**Note:** If you set DDN as the national personality, DDN address translation is enabled automatically and this parameter has no effect.

**Syntax:**

**disable** ddn-address-translations

**Note:** Disabling ddn-address-translations is no longer allowed. This feature defaults to enabled when the national personality selected is DDN, and defaults to disabled in all other cases.

incoming-calls-barred

lower-dtr

outgoing-calls-barred

## National Enable

Use the **national enable** command to enable a feature defined in the National Personality configuration.

## Configuring the X.25 Network Interface

**Syntax:**

**national enable**

accept-reverse-charges
bi-cug
bi-cug-outgoing-access
cug
cug-deletion
cug-incoming-access
cug-insertion
cug-outgoing-access
cug-zero-override
flow-control-negotiation
frame-ext-seq-mode (required for X.31)
packet-ext-seq-mode
request-reverse-charges
suppress-calling-addresses
throughput-class-negotiation
truncate-called-addresses

**accept-reverse-charges**
Accepts reverse charge calls during call establishment. This option is not available for DDN.

**DDN Default**
off

**GTE Default**
on

**bi-cug** Enables the bilateral closed user group facility on this device. By default, this facility is disabled.

**Note:** You cannot add any bilateral CUGs unless this parameter is enabled.

**bi-cug-outgoing-access**
Enables the bilateral CUG with outgoing access facility on this device. By default, this facility is disabled.

**cug** Enables the closed user group facility on this device. By default, this facility is disabled.

**Note:** You cannot add any CUGs unless this parameter is enabled.

**cug-deletion**
Deletes a CUG facility from a call packet received from XTP before transmitting it over X.25. By default, this function is disabled.

**cug-incoming-access**
Enables the CUG with incoming access facility on this device. By default, this facility is disabled.

**cug-insertion**
Inserts the appropriate (address-specific, protocol-specific, or

interface-specific) preferred cug number into a call request received by XTP from the X.25 interface before transmitting the request over IP. If there is already a CUG facility in the call packet, it will not be replaced. By default, this function is disabled.

**cug-outgoing-access**
Enables the CUG with outgoing access facility on this device. By default, this facility is disabled.

**cug-zero-override**
Causes the closed user group facility to ignore any CUG facility in call request packets with a CUG number of 0. By default, this function is disabled.

**flow-control-negotiation**
Enables the negotiation of packet and window size during call setup of SVCs.

**DDN Default**
on

**GTE Default**
on

**frame-ext-seq-mode**
Sets the frame layer sequence numbering to modulo 128 (i.e., 0 through 127).

**DDN Default**
off (must be on for X.31)

**GTE Default**
off

**packet-ext-seq-mode**
Enables the packet layer to use extended sequence numbers (0 through 127).

**DDN Default**
off

**GTE Default**
off

**request-reverse-charges**
Requests reverse charges for all outgoing calls.

**DDN Default**
off

**GTE Default**
on

**suppress-calling-address**
Suppresses the source address in call packets.

**DDN Default**
off

**GTE Default**
off

**throughput-class-negotiation**
Enables the registration of throughput class.

**DDN Default**
off

**GTE Default**
on

**truncate-called-addresses**
Enables truncation of the called DTE address when transmitting a call to a DTE. This option applies only to XTP circuits.

**DDN Default**
off

**GTE Default**
off

## National Disable

Use the **national disable** command to disable a feature defined by the National Personality configuration.

**Syntax:**

**national disable**

accept-reverse-charges
bi-cug
bi-cug-outgoing-access
cug
cug-deletion
cug-incoming-access
cug-insertion
cug-outgoing-access
cug-zero-override
flow-control-negotiation
frame-ext-seq-mode
packet-ext-seq-mode
request-reverse-charges
suppress-calling-addresses
throughput-class-negotiation
truncate-called-addresses

## 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

n2-timeouts

packet-size . . .

reset . . .

restart . . .

min-recall

min-connect

collision-timer

standard-version

t1-timer

t2-timer

truncate-called-addr-size

**call-req**

Specifies the number of 10-second intervals permitted before giving up on a call request and clearing it. A zero indicates an infinite wait. In a list command output, this is displayed as the t21 timer.

**DDN Default**

20 decaseconds

**GTE Default**

20 decaseconds

**clear-req** *retries* **OR** *timer*

Specifies the number of clear request retransmissions.

**Retries**

Number of clear request transmissions permitted before action is taken. In a list command output, this is displayed as the r23 retry count.

**DDN Default**

retries=1

**GTE Default**

retries=1

**Timer** Number of 10–second intervals to wait before retransmitting a clear request packet. A zero in the timer value indicates an indefinite wait. In a list command output, this is displayed as the t23 timer.

**DDN Default**

18 decaseconds

**GTE Default**

18 decaseconds

**disconnect-procedure** *passive* **OR** *active*

Specifies the type of disconnect procedure to use when disconnecting.

**DDN Default**

passive

**GTE Default**
passive

**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

**frame-window-size**
Specifies the number of frames that can be outstanding before acknowledgment.

**DDN Default**
7

**GTE Default**
7

**n2-timeouts**
Specifies the number of times the retransmit timer (T1) can expire before the interface is recycled.

**DDN Default**
20

**GTE Default**
20

**packet-size** *default* **OR** *maximum* **OR** *window*
Specifies the size of the packet.

**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 acknowledgment 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. In a list command output, this is displayed as the r22 retry count.

**DDN Default**
1

**GTE Default**
1

**timer** Number of 10–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. In a list command output, this is displayed as the t22 timer.

**DDN Default**
18 decaseconds

**GTE Default**
18 decaseconds

**restart** *retries* **OR** *timer*
Specifies the number of restart request transmissions.

**retries**
Number of restart request transmissions permitted before the interface is recycled. The range is 0 to 255. In a list command output, this is displayed as the r20 retry count.

**DDN Default**
1

**GTE Default**
1

**timer** Number of 10–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. In a list command output, this is displayed as the t20 timer.

**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 to 255 seconds.

**DDN Default**
10 seconds

**GTE Default**
10 seconds

**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 to 255 seconds.

**DDN Default**
90 seconds

**GTE Default**
90 seconds

**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 to 255 seconds.

**DDN Default**
10 seconds

**GTE Default**
10 seconds

**standard-version**
Options are none, v1980, v1984, and v1988.

**DDN Default**
1984

**GTE Default**
1984

**t1-timer**
Specifies the frame retransmit time in seconds. The range is 1 to 255.

**DDN Default**
4 seconds

**GTE Default**
4 seconds

**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 to 255.

**DDN Default**
0

**GTE Default**
0

**truncate-called-addr-size**
Specifies the number of characters truncated from the end of a called address. This parameter pertains only to XTP circuits. The range is 0 to 10.

**DDN Default**
2

**GTE Default**
2

## 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.

**Syntax:**

**national restore**

- all
- accept-reverse-charges
- bi-cug
- bi-cug-outgoing-access
- call-req
- clear-req . . .
- cug
- cug-deletion
- cug-incoming-access
- cug-insertion
- cug-outgoing-access
- cug-zero-override
- disconnect-procedure . . .
- dp-timer
- flow-control-negotiation
- frame-ext-seq-mode
- frame-window-size
- min-collission-timer
- min-connect-timer
- min-recall-timer
- network-type . . .
- n2-timeouts
- packet-size . . .
- packet-ext-seq-mode
- request-reverse-charges
- reset . . .
- restart . . .
- standard-version
- suppress-calling-addresses
- throughput-class-negotiation
- t1-timer

t2-timer

truncate-called-addresses

truncate-called-addr–size

# Add

Use the **add** command to add an X.121 address, a DDN X.25 Address, a protocol configuration, or a PVC definition.

**Syntax:**

**add** address
bi-cugs
cugs
htf-address
protocol
pvc

**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 mapping of a protocol address and the X.121 address must be unique unless the protocol is APPN or DLSw. A protocol address cannot map to more than one X.121 address. Also, a specific X.121 address cannot map to more than one protocol address. The **set address** command is used to set the local X.25 address. After setting the local X.25 address, you can use an X.25 remote address to dial out and an optional incoming remote address for call ID. IF only remote called address is entered, then this address will be used for outgoing calls and incoming call verification.

**Example: `add address`**

**IP example:**

```
Protocol [IP]?  IP
IP Address [0.0.0.0]? 128.185.1.2
Enc Priority 1 []? CC
Enc Priority 2 []? SNAP
Enc Priority 3 []? Null
X.25 Address []? 1234590
Remote address []?
Pref CUG  []? 11
CUG (2)  []? 12
CUG (3)  []? 13
CUG (4)  []? 14
CUG (5)  []? 15
Pref BI-CUG  []? 21
BI-CUG (2)  []? 22
BI-CUG (3)  []?
```

**IPX example:**

```
Protocol [IP]?  IPX
CUD Field Usage (Standard or Proprietary)
IPX Host Number (in hex) []?
Enc Priority 1 []? SNAP
Enc Priority 2 []?Null
X.25 Address []?
```

```
Pref CUG [] ?
Pref Bi-CUG[]? 1
BI-CUG (2)[]? 3
BI-CUG (3)[]
```

**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, SNAP, or Null. For IPX, valid choice is SNAP or Null.

**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.

**pref cug**
Specifies the preferred closed user group number for this DTE. The DTE uses this CUG when placing outgoing calls.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the closed user group facility using the **national enable** command.

**CUG** Specifies the closed user group numbers for this DTE. Up to five CUGs may be defined, including the pref CUG.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the closed user group facility using the **national enable** command.

**pref bi-cug**
Specifies the bilateral closed user group number for this DTE. The DTE uses this CUG when placing outgoing calls.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the bilateral closed user group facility using the **national enable** command.

**bi-cug** Specifies the bilateral closed user group numbers for this DTE. Up to five CUGs may be defined.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the bilateral closed user group facility using the **national enable** command.

**cugs** Specifies the closed user group number for this X.25 interface.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the closed user group facility using the **national enable** command.

**Example:**

```
add cugs
Pref CUG  []? 23
CUG (2)  []? 24
CUG (3)  []? 25
CUG (4)  []? 26
CUG (5)  []? 27
```

**pref cug**
Specifies the preferred closed user group number for this DTE. This DTE uses this CUG when placing outgoing calls.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the closed user group facility using the **national enable** command.

**cug** Specifies the closed user group numbers for this DTE. Up to five CUGs may be defined.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the closed user group facility using the **national enable** command.

**bi-cugs**
Specifies the closed user group number for this DTE.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the closed user group facility using the **national enable** command.

**Example:**

```
add bi-cugs
Pref BI-CUG  []? 23
BI-CUG (2)  []? 24
BI-CUG (3)  []? 25
BI-CUG (4)  []? 26
BI-CUG (5)  []? 27
```

**pref bi-cug**

Specifies the preferred closed user group number for this DTE. This DTE uses this BI-CUG when placing outgoing calls.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the bilateral closed user group facility using the **national enable** command.

**bi-cug** Specifies the closed user group numbers for this DTE. Up to five BI-CUGs may be defined.

**Valid values:** 0 to 9999

**Default value:** None

**Note:** You will not be prompted for this value if you have not enabled the bilateral closed user group facility using the **national enable** command.

**htf-address**

Adds a Defense Data Network (DDN) X.25 address translation.

**Example:**

```
add htf-address
Protocol [IP]
Convert HTF address
```

**Protocol**

Specifies the protocol that you are running over the X.25 interface. DDN supports IP only.

**Convert HTF address**

Converts the protocol address to a 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]?
Max VCs [4]?
Pref CUG  []? 1
CUG (2)  []? 2
CUG (3)  []? 3
CUG (4)  []? 4
CUG (5)  []? 5
Pref BI-CUG  ]? 11
```

```
BI-CUG (2)  []? 12
BI-CUG (3)  []? 13
BI-CUG (4)  []? 14
BI-CUG (5)  []? 15
```

QLLC example:

```
X.25 Config> add prot
Protocol [IP]? dls
Idle timer [30]?
QLLC response timer (in decaseconds) [2]?
QLLC response count [3]?
Accept Reverse Charges [N]?
Request Reverse Charges [N]?
Station Type (1) PRI  (2) SEC  (3) (PEER) [3]?
Max Packet Size [128]?
Packet window size [7]?
Max Message Size [1500]?
Call User Data (in hex, 0 for null) []?
Pref CUG  []? 20
CUG (2)  []? 21
CUG (3)  []?
Pref BI-CUG  []?
```

**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 256 bytes. The maximum value that can 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.

**pref CUG, CUG, pref bi-cug, bi-cug**
See **add address** command.

**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.

**Remote Address**
Specifies the remote address for caller ID on received calls.

**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. The maximum for X.31 is 256 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
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:**

**delete** address
bi-cugs
cugs
protocol . . .
pvc

**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]?
```

**bi-cugs**
Deletes a bilateral closed user group number used by this interface.

**Valid values:**

**Y** Deletes the current CUG.

**N** Does not delete the current CUG.

**ALL** Deletes all remaining CUGs.

**Q** Stops deleting any remaining CUGs.

**Example:**

```
delete bi-cugs
Delete Pref BI-CUG [Y]?
Delete BI-CUG (2) [Y]? N
Delete BI-CUG (3) [Y]? q
```

**cugs** Deletes the closed user group numbers used by this interface. This command works similar to the **delete bi-cug** command.

**Example:**

```
del cug

Delete Pref CUG [Y]?
Delete CUG (2) [Y]?
Delete CUG (3) [Y]? q
```

**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.

**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:**

**list** address
all
cugs
detailed
protocols
pvc
summary

**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
            CUGS: 11 12 13 14 15         BI-CUGS: 21 22
```

**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

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
Truncate Called Addresses:  off
Number of digits to truncate called addresses to:  2
CUG Support: off               BI-CUG Support: off
CUG Outgoing Access: off       CUG Incoming Access : off
BI-CUG Outgoing Access: off    CUG 0 Override: off
CUG Isertion: off              CUG deletion: 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
```

**cugs** Lists the CUG and BI-CUG numbers for each X.25 interface in this device.

**Example:**

```
li cugs
CUGS: 23 24 25 26 27
```

**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 detail

X.25 National Personality Configuration

Follow CCITT: on        OSI 1984:     on        OSI 1988:       off
Request Reverse Charges:  off   Accept Reverse Charges:     off
Frame Extended seq mode:  off   Packet Extended seq mode:   off
Incoming Calls Barred:    off   Outgoing Calls Barred:      off
Throughput Negotiation:   on    Flow Control Negotiation:   off
Suppress Calling Addresses: off DDN Address Translation:    off
Truncate Called Addresses:  off
Number of digits to truncate called address to:  2
CUG Support: off                BI-CUG Support: off
CUG Outgoing Access: off        CUG Incoming Access : off
BI-CUG Outgoing Access: off     CUG 0 Override: off
CUG Isertion: off               CUG deletion: off
T21 (Call Request Timer):     20 decaseconds
T23 (Clear Request Timer):    18 decaseconds (1 retries)
T22 (Reset Request Timer):    18 decaseconds (1 retries)
T20 (Restart Request Timer):  18 decaseconds (1 retries)
Min Recall Timer:       10 seconds
Min Connect Timer:      90 seconds
Collision Timer:        8 seconds
T1 Timer:  4.00 seconds     N2 timeouts: 20
T2 Timer:  0.00 seconds     DP Timer:    500 milliseconds
Standard Version:     1984      Network Type: CCITT
Disconnect Procedure: active
Window Size     Frame:   7      Packet:  2
Packet Size     Default: 256    Maximum: 256
```

**protocols**

Lists all the defined protocol configurations. See “Add” on page 412 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
              CUGS: 11 12 13 14 15         BI-CUGS: 21 22
```

```
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 ]
          CUGS: 11 12 13 14 15      BI-CUGS: 21 22
```

**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
```

## Accessing the Interface Monitoring Process

To monitor information related to the X.25 network interface, access the interface monitoring process as follows:

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 “Configuration” on page 102 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 monitoring prompt is displayed on the console. You can then view information about the X.25 interface by entering the X.25 monitoring commands.

# X.25 Monitoring Commands

This section summarizes and explains all the X.25 monitoring commands. The X.25 monitoring commands allow you to view the parameters and statistics of the interfaces and networks that transmit X.25 packets. Monitoring 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 monitoring commands at the `X.25>` prompt. Table 58 shows the commands.

*Table 58. X.25 Monitoring Command Summary*

| Monitoring Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to display the current active PVCs and SVCs.

**Syntax:**

**list** pvcs
svcs

**pvc** Displays the configured permanent virtual circuits.

**svc** Displays the active switched virtual circuits.

**Example:**

```
list svc

LCN/  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** all
frame

packet

physical

**all** Displays the parameters for the packet, frame, and physical levels.

**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: Inhibit Idle RRs No   MOD 128 NO   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
T20 (Restart)       = 18  R20 (Retry)         =     1
T21 (Call)         =     20
T22 (Reset)         = 18  R22 (Retry)         =     1
T23 (Clear)         = 18  R23 (Retry)         =     1
Network Type        = GTE  Equipment Type     =   DTE
```

**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:**

**statistics**

all

frame

packet

physical

**all** Displays the statistics for the packet, frame, and physical levels.

**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
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

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
Output frame counters:
  DMA/FIFO underrun errors       0   Output aborts sent          0
```

## X.25 Network Interfaces and the GWCON Interface Command

While X.25 interfaces have their own monitoring 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 10. The Operating/Monitoring Process (GWCON - Talk 5) 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 list 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 37. Using XTP

This chapter describes the X.25 Transport Protocol (XTP) for transporting X.25 traffic over TCP/IP. Included are the following sections:



## The X.25 Transport Protocol

X.25 Transport Protocol (XTP) provides you with the services of a "protocol forwarder." A protocol forwarder is the focal point for inbound and outbound protocol packet processing. Forwarders receive packets on one network interface and send them to another interface.

XTP is designed to work with X.25 devices that are situated at multiple remote sites. In such environments, XTP can eliminate the use of X.25 packet-switched networks for communicating with servers at one or more centralized locations.

To enable this, you use routers at the server and remote locations to encapsulate the data and deliver the X.25 packets between the clients and server via TCP/IP.

Figure 37 on page 432 illustrates a network configuration before and after using XTP.

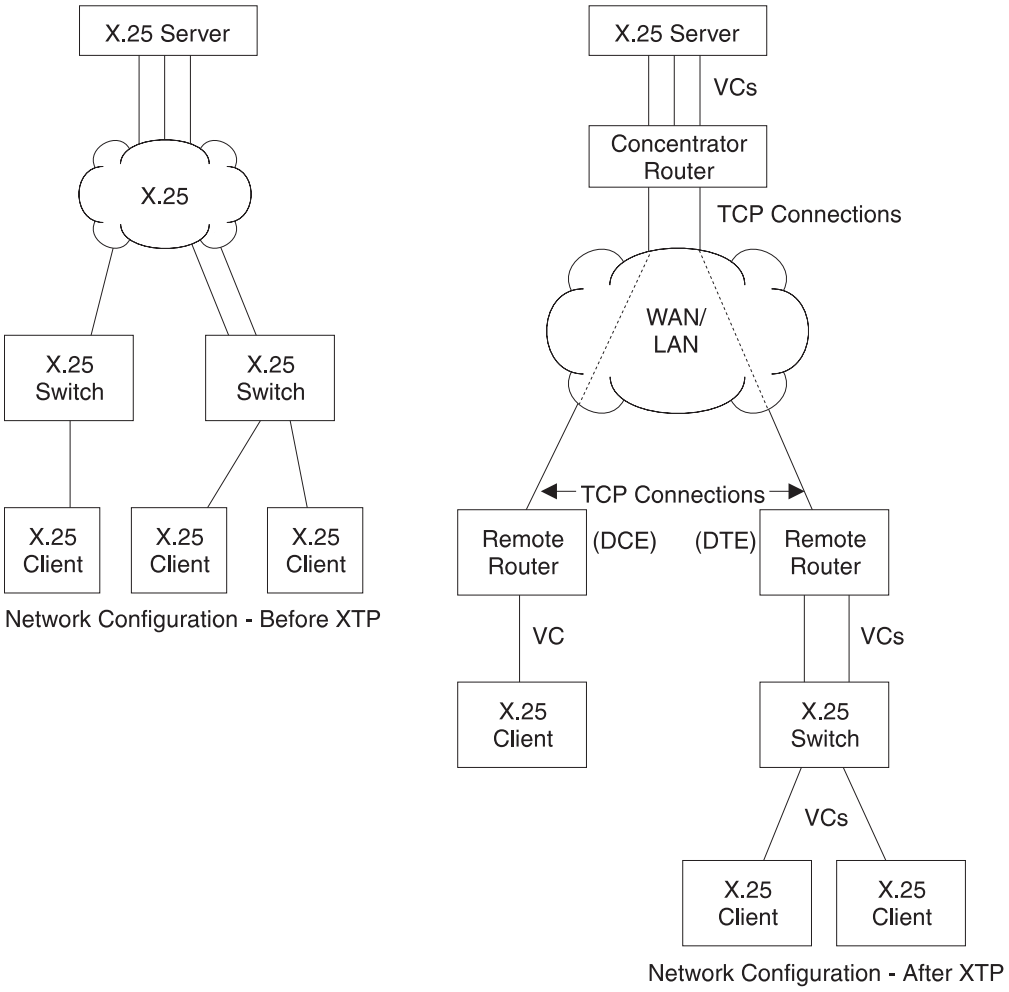



*Figure 37. Configuration Before and After XTP*

## Configuration Information

X.25 recognizes an incoming call for XTP based on the node addresses configured for XTP. Therefore, in order to transport X.25 traffic between the X.25 nodes, you must configure X.25 to map to the data terminal equipment (DTE) address and IP addresses of the routers to which the nodes are connected.

For example, in Figure 37, you configure X.25 clients on remote routers and on the concentrator router. *Remote routers* in this example are the routers that connect the X.25 clients to the TCP/IP network that is used to access the X.25 server; the *concentrator router* connects the X.25 server to the TCP/IP network that is used to access the remote routers.

**Note:** When you configure XTP, if a router is connected to an X.25 switch, it is considered to be DTE. If it is not connected to a switch, it is considered to be DCE (Data Circuit-Terminating Equipment).

To configure a router for XTP, define the following information from the `XTP config>` prompt and then restart the router:

- Local DTEs
- Peer routers

- Remote DTEs
- PVCs
- CUGs

**Local DTEs**
X.25 nodes connected to the X.25 interfaces on the router

To configure local DTEs, use the X.121 address that is assigned to the local DTE. Multiple local DTEs can be configured on an interface.

**Peer Routers**
Routers with which you communicate over TCP/IP

Peer routers can differ depending on "point of view". For example, in Figure 37 on page 432, the *two remote routers* are the peer routers from the perspective of the concentrator router. However, the *concentrator router* is the peer router from the perspective of the two remote routers.

You designate the peer router by its internal IP address.

**Remote DTEs**
Remote X.25 nodes to which the local X.25 nodes open connections and exchange data. Use the X.121 address that is assigned to the remote DTE.

Configure a *unique* IP address for each peer router. For example, in Figure 37 on page 432, the concentrator router must know the unique IP address of each remote router, and each remote router must know the IP address of the concentrator router.

**PVC** A permanent channel that remains connected after X.25 restarts.

PVCs, because they are constant channels, are similar to leased telephone lines. A PVC, in the XTP context, is a PVC from a local X.25 DTE node to a remote X.25 DTE.

When you configure a router for PVCs, map the IP address of the peer router and the PVC number of the remote and local DTE. A PVC is identified by four pieces of information which are the:

- Logical channel number of the local PVC
- X.121 address of the local DTE
- Logical channel number of the PVC on the remote (peer) router
- X.121 address of the remote DTE

**CUGS** The closed user groups for the XTP protocol. See "Understanding Closed User Groups" on page 393.

Additional configuration information can be found at "Configuring XTP" on page 436 and at "XTP Configuring Commands" on page 445.

## DTE Address Wildcards

The "*" wildcard is available for DTE address configuration. This is in addition to the "?" character that can be specified in a DTE address to represent any one digit in that position in the address. For example, a specification of "1?2?3" can match address 18243 where the first, third, and fifth digits are 1, 2, and 3, respectively.

The "*" wildcard character can represent any string of zero or more digits. Its use is limited to the end of a DTE address specification. For example: "123*", "5555*", "9*"

or "*". The special case of a DTE address of "*" represents any DTE address, even a null address. The null address is useful for handling incoming calls with no calling address in the X.25 Call Request packet.

Use of the "*" wildcard increases the chances for adding a local or a remote DTE address that conflicts with an existing address. The **add local-dte** and **add remote-dte** commands are enhanced to provide the conflicting address when the user attempts to add a DTE address that conflicts with an existing address.

**Example:**
```
xtp config> add local-dte
Interface number [0]? 1
DTE address [ ] 123456
DTE address [ ]?

XTP config>add local-dte
Interface number [0]?1
DTE address [ ]?1*
DTE address conflicts with existing DTE address 123456
```

## XTP Backup Peer Function

The Backup Peer Function allows the association of multiple peer routers with a remote DTE. The user specifies a list of peer routers associated with a remote DTE.

Example:

```
XTP config>add rem
DTE address [ ]?123456
Peer router's internal IP Address [0.0.0.0]?10.0.0.2
Peer router's internal IP Address [0.0.0.0]?10.0.0.4
Peer router's internal IP Address [0.0.0.0]?11.0.0.1
Peer router's internal IP Address [0.0.0.0]?
```

When an incoming call for the remote DTE is received, a connection is attempted through each router in the list in the same order that they appear for the remote DTE.

## Searching for a Remote DTE

When a DTE initiates a call for a remote DTE, both DTE addresses are inspected to determine if they are acceptable for X.25 transport. If they are acceptable, the X.25 Transport protocol forwarder determines through which peer router to attempt to complete the call. It starts with the first router in the remote DTE's list of peer routers in its search. The first condition that must be met is an active TCP connection to the peer router. If there is not an active TCP connection to the peer, the next router in the list is checked. When an active TCP connection is found, an attempt is made to complete the call. The Connection Request Timer is started to time the call connection process.

The remote DTE search is terminated by one of the following events:

- Successful completion of the call through the peer router

  This completes call setup processing and ends the search for the remote DTE.
- Rejection of the call by the peer router

  This causes the search for the remote DTE to proceed to the next router in the peer router list.
- Expiration of the Connection Request Timer

This causes the search for the remote DTE to proceed to the next router in the peer router list.

If a pass through the list of peer routers is completed without a successful connection through any of the peer routers, the call to the local DTE is cleared.

## Connection Request Timer

The Connection Request Timer is used to ensure that no call setup procedure hangs for an indeterminable time. There is a timer configured for each peer router.

Example:

```
XTP config>add peer-router
Router's internal IP Address [0.0.0.0]?10.0.0.2
Connection setup timeout [230]?60
```

The Connection Request Timer can be configured from 10 to 480 seconds. The default is 230 seconds. This default was determined based on the fact that the default setting for the X.25 Call Request Timer is 200 seconds.

The timer is started when an attempt is made to complete a call through a peer router. It is stopped when the call attempt is either accepted or rejected by the peer router.

## Local XTP

Local XTP allows you to route incoming X.25 traffic to the same or different interfaces on the current router. To configure local XTP, specify the router's internal IP address as a peer address on the **add peer** command.

## XTP and Closed User Groups

XTP supports closed user groups through the local DTE address defined by the **add local** or the **add cug** command. To enable XTP to use closed user groups, you must:

- Enable CUG or BI-CUG on the appropriate X.25 interfaces.
- Supply the XTP protocol-specific CUGs using the **add cug** and **add bi-cug** commands, if desired.
- Supply the appropriate closed user group numbers in the **add local** command. These include:
  - Closed user group number
  - Preferred closed user group number
  - Bilateral closed user group number
  - Preferred bilateral closed user group number
- Enable CUG insertion or deletion for the interface in the **national enable cug_insertion** or **national enable cug_deletion** commands, if desired.
- Enable the CUG 0 override option on the **national enable cug 0 override** command, if desired.

# Configuring XTP

XTP is a protocol forwarder used to transport X.25 traffic over TCP/IP. XTP allows existing X.25 devices to communicate over a TCP/IP backbone and migrate from an X.25 network to a network of your choice.

## Configuration Procedures

This section defines the detail for configuring the network displayed in Figure 38.

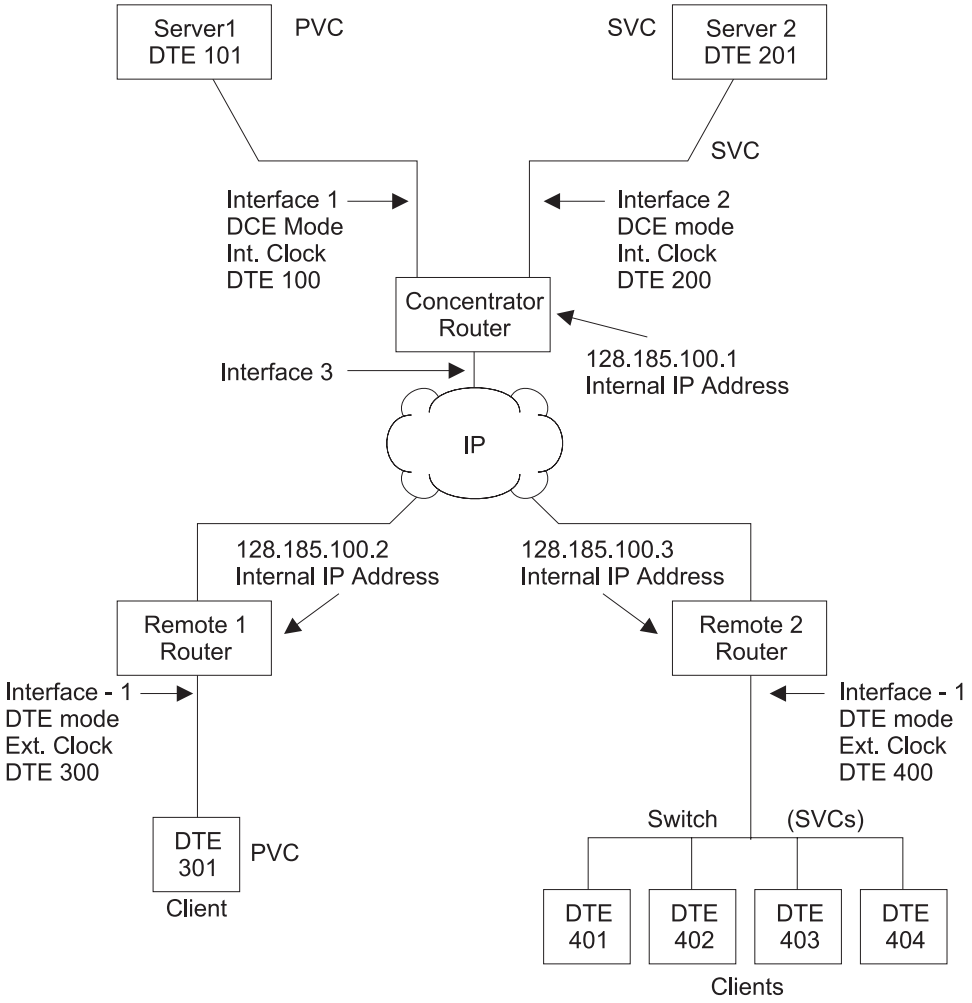



*Figure 38. Sample XTP Configuration*

This configuration shows three routers, the Concentrator router, Remote 1 router, and Remote 2 router. To make XTP operational on this network, perform the following steps for each of these routers:

- Set the data link
- Configure the IP interface
- Configure X.25
- Set the National Personality values
- Define the IP address

- Set the Internal IP address
- Configure XTP

**Note:** New configurations do not take effect until you restart the router.

## Setting the Data Link

The data link defines the protocol you are using to send data packets over the network. Define the data link between the router you are configuring and each serial interface. The example in Figure 38 on page 436 configures a concentrator router with three serial interfaces, two for X.25 and one for PPP.

Set the data-link protocol for the serial interfaces:

```
Config>set data-link X25 1
Config>set data-link x25 2
Config>set data-link ppp 3
```

## Configuring the IP Interface

In Figure 38 on page 436, the IP interface is PPP; enter **network 3** at the `Config>` prompt to configure this PPP interface:

```
Config>network 3
PPP interface configuration
```

**Note:** This procedure does not include details about the configuration of PPP. For details, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*

## Configuring X.25

Before configuring XTP, configure the X.25 parameters for each interface. The following example configures the basic parameters for X.25 and is based on the topology in Figure 38 on page 436.

The parameters you need to configure depend on your network topology. For details about all the X.25 parameters, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*

### Interface 1

Use the following instructions to configure *Interface 1* on the concentrator router as defined in Figure 38 on page 436.

1. At the `Config>` prompt, enter **network** followed by the number of the X.25 interface. In this example, it is interface 1.

   ```
   Config>network 1
   X.25 User Configuration
   X.25 Config>
   ```

2. Add the XTP protocol to the X.25 interface and define general interface values. Enter **add protocol xtp** at the X.25 `Config>` prompt. This command needs to be entered *one time only*.

   ```
   X.25 Config>add protocol xtp
   Window Size [2]?
   Default Packet Size [128]?
   Maximum Packet Size [256]?
   ```

3. Specify the network address by entering **set address** X.25 node address. In Figure 38 on page 436, the node address (DTE address) is 100.

```
X.25 Config>set address 100
```

4. Enter **set clocking** followed by **internal** or **external** based on your router type.

```
X.25 Config>set clocking internal
```

5. Enter **set speed** followed by the access rate (line speed).

```
X.25 Config>set speed
Access rate in bps [9600]?19200
```

6. Enter **set equipment-type** and specify whether the frame and packet levels act as DCE or DTE.

```
X.25 Config>set equipment-type dce
```

7. Enter **set pvc** and define the lowest and the highest PVC you are using.

```
X.25 Config>set pvc low 1
X.25 Config>set pvc high 1
```

8. Enter **add pvc** to define individual PVCs.

```
X.25 Config>add pvc
Protocol [IP]?xtp
Packet Channel [1]?
Destination X.25 Address [ ]?101
Window Size [2]?
Packet Size [128]
```

9. (Optional) Enter **national enable truncate-called-addresses**. If you want to truncate the called address size, enter **national set truncate-called-addr-size** followed by the number of digits to truncate the called DTE address to.
10. (Optional) Enable CUG support, CUG insertion, and CUG deletion as required.

## Interface 2

Use the following instructions to configure interface 2.

1. At the Config> prompt, enter **network** followed by the number of the X.25 interface. In Figure 38 on page 436, it is 2.

```
Config>network 2
X.25 User Configuration
X.25 Config>
```

2. Use the same procedures as defined in “Interface 1” on page 437 to set the following parameters for interface 2:
   - address = 200
   - clocking = internal
   - speed = 19200
   - equipment = dce
3. Enter **set svc** and define the lowest and highest SVC you are using.

   There are three types of SVCs: two-way, inbound and outbound. The defaults are “svc low-two-way = 1” and “svc high-two-way = 64.” All other SVC types default to 0. For additional information on SVCs and PVCs, refer to *Software User’s Guide for Nways Multiprotocol Access Services Version 3 Release 1*

```
X.25 Config>set svc ?
X.25 Config>set svc low-inbound 0
X.25 Config>set svc high-inbound 0
X.25 Config>set svc low outbound 0
X.25 Config>set svc high outbound 0
X.25 Config>set svc low-two-way 2
X.25 Config>set svc high-two-way 2
```

4. Exit the X.25 Config> prompt.

```
X25 Config>exit
Config>
```

## Setting the National Personality

Each X.25 public network has its own standard configuration. The National Personality refers to a group of 28 variables that define the characteristics of the public data network. These variables provide the router with control information for packets transferred over the link and influence the X.25 facilities used between and XTP router and its local DTE.

All facilities contained in incoming call requests are passed on to the peer router, regardless of whether the local router was configured to support that facility. For example, when packet size negotiation is requested in the incoming call and flow control negotiation is not configured in the router.

The router will insure any packet size and window size being negotiated is within the range specified when defining the X.25 interface. For example, a packet window greater than 7 is negotiated down to 7 if packet-ext-seq-mode has not been defined for the X.25 interface.

To view the configuration values, enter **list detailed** at the `X.25 Config>` prompt. To set the default values for the national personality, enter **set national-personality** at the `X.25 Config>` prompt. For further information, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*

## Defining the IP Address

Before you configure the Concentrator router (as displayed in Figure 38 on page 436 ) for XTP, define the IP address for this router. Enter **protocol ip** at the `Config>` prompt and enter **add address** at the `IP config>` prompt.

```
Config>protocol ip
IP config>add address
Which net is this address for [0]?3
New address [0.0.0.0]?128.185.100.7
Address mask [255.255.0.0]?255.255.255.0
```

## Setting the Internal IP Address

Each router identifies its peer routers by the internal IP address of the peer routers.

To set the internal IP address of the peer router, enter **set internal IP address** at the `IP Config>` prompt.

```
IP config>set internal-ip-address
Internal IP address [0.0.0.0]?128.185.100.1
```

## Configuring XTP

After you have configured X.25 and defined the IP address, you are ready to configure XTP for the router.

If you need further configuration information when configuring XTP, see "XTP Configuring Commands" on page 445.

**Note:** When configuring your network for XTP, remember that the peer routers are always the routers you are communicating with over TCP/IP. Therefore, the peer router can differ depending on the point of view. When configuring the

routers defined as Remote 1 router and Remote 2 router in Figure 38 on page 436 , to them the peer router is the Concentrator router.

Implement the following steps to configure XTP for the router:

1. To access the `XTP config>` prompt, enter **protocol xtp** at the `Config>` prompt.
2. Add interface 1 to the XTP configuration. Enter **add local-dte** at the `XTP Config>` prompt.

```
XTP config>add local-dte
Interface number [0]?1
Allow inbound calls without calling DTE address? (Y or N) [N]? n
DTE address [ ]?101
Pref CUG []? 18
CUG (2) []? 2
CUG (3) []?
Pref BI-CUG [0]?
DTE address [ ]?
```

   Entering a null DTE address ends the command input.
3. Add interface 2 to the XTP configuration. Enter **add local-dte** at the `XTP Config>` prompt.

```
XTP config>add local-dte
Interface number [0]?2
Allow inbound calls without calling DTE address? (Y or N) [N]? n
DTE address [ ]?201
DTE address [ ]?
```

   Entering a null DTE address ends the command input.
4. (Optional) Add XTP protocol-specific CUGs.

```
add cug
        Pref CUG  []? 11
        CUG (2)  []? 12
        CUG (3)  []? 13
        CUG (4)  []? 14
        CUG (5)  []? 15
add bi-cug
        Pref BI-CUG  []? 21
        BI-CUG (2)  []? 22
        BI-CUG (3)  []?
```

5. Add Remote 1 router as the peer router. Enter **add peer-router** and enter the IP address of this router.

```
XTP config>add peer-router
Router's internal IP Address [0.0.0.0]?128.185.100.2
Connection setup timeout [230]?
```

6. Add the remote DTE for Remote 1 router. Enter **add remote-dte** and enter the IP and DTE address of this DTE.

```
XTP config>add remote-dte
DTE address [ ]?301
Peer router's internal IP Address [0.0.0.0]?128.185.100.2
Peer router's internal IP Address [0.0.0.0]?
```

   **Note:** A remote DTE is *required* only if one of the following applies:
   - The Concentrator Router will be initiating XTP connections to the remote DTE due to incoming calls from its local DTEs.
   - The DTE is part of an XTP PVC definition.
7. Add Remote 2 router (as the peer router). Enter **add peer-router** and enter the IP address of this router.

```
XTP config>add peer-router
Router's internal IP Address [0.0.0.0]?128.185.100.3
Connection setup timeout [230]?
```

8. Add the remote DTEs for Remote 2 router. Enter add remote-dte and enter the IP and DTE addresses of this DTE.

```
XTP config>add remote-dte
DTE address [ ]?401
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?

XTP config>add remote-dte
DTE address [ ]?402
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?

XTP config>add remote-dte
DTE address [ ]?403
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?

XTP config>add remote-dte
DTE address [ ]?404
Peer router's internal IP Address [0.0.0.0]?128.185.100.3
Peer router's internal IP Address [0.0.0.0]?
```

9. Add an XTP PVC to logically associate the local PVC to Server 1 with the remote DTE 301.

```
XTP config>add pvc
Local PVC number [1]? 1
Local X.25 DTE address [ ]? 101
Remote PVC number [1]? 1
Remote X.25 DTE address [ ]?301
```

When entering DTE addresses, you can specify either of the following:

A '?' in place of any digit. The '?' means any single digit in this digit position.

An '*' as the last digit of an address to represent any combination of zero or more digits.

## Sample Configuration of Remote Routers

The following is a sample configuration of Remote 1 router and Remote 2 router (see Figure 38 on page 436). The process is the same as that defined in the section at "Configuration Procedures" on page 436.

### Remote 1 router

```
*talk 6

Config>set data-link x25 1
Config>set data-link ppp 2
Config>network 1

X.25 Config>set address 300
X.25 Config>set clocking internal
X.25 Config>set speed 19200
X.25 Config>set equipment-type dce
X.25 Config>set pvc low 1
X.25 Config>set pvc high 1
X.25 Config>add pvc
Protocol [IP]?xtp
Packet Channel [1]?1
Destination X.25 Address [ ]?301

Window Size [2]?
Packet Size [128]?
X.25 Config>exit
Config>

Config>protocol ip
IP config>add address
Which net is this address for [0]?2
New address [0.0.0.0]?128.185.100.8
Address mask [255.255.0.0]?255.255.255.0

IP config>set internal-ip-address
Internal IP address [0.0.0.0]?128.185.100.2
IP Config>exit
Config>
```

```
Config>protocol xtp
XTP config>add local-dte
Interface number [0]?1
Allow inbound calls without calling DTE address? (Y or N) [N]? n
DTE address [ ]?301
DTE address [ ]?

XTP config>add peer-router
Router's IP address?128.185.100.1

XTP config>add remote-dte
DTE address [ ]?101
Peer router's internal IP Address ]0.0.0.0]?128.185.100.1
Peer router's internal IP Address [0.0.0.0]?

XTP config>add pvc
Local PVC number [1]? 1
Local X.25 DTE address [ ]? 101
Remote PVC number [1]? 1
Remote X.25 DTE address [ ]?301
```

## Remote 2 router

```
*talk 6

Config>set data-link x25 1
Config>set data-link ppp 2
Config>network 1

X.25 Config>set address 400
X.25 Config>set clocking external
X.25 Config>set speed 19200
X.25 Config>set equipment-type dte
X.25 Config>set svc low-inbound 0
X.25 Config>set svc high-inbound 0
X.25 Config>set svc low-outbound 0
X.25 Config>set svc high-outbound 0
X.25 Config>set svc low-two-way 1
X.25 Config>set svc high-two-way 64
X.25 Config>add protocol
Protocol [IP]?xtp
Window Size [2]?
Default Packet Size [128]?
Maximum Packet Size [256]?
X.25 Config>exit

Config>protocol ip
IP config>add address
Which net is this address for [0]?2
New address [0.0.0.0]?128.185.100.9
Address mask [255.255.0.0]?255.255.255.0

IP config>set internal-ip-address
Internal IP address [0.0.0.0]?128.185.100.3
IP Config>exit
Config>

Config>protocol xtp
XTP config>add local-dte
Interface number [0]?1
Allow inbound calls without calling DTE address? (Y or N) [N]? n
DTE address [ ]?401
Pref CUG  []? 23
CUG (2)  []? 24
CUG (3)  []? 25
CUG (4)  []? 26
CUG (5)  []? 27

DTE address [ ]?402
Pref CUG  []?
DTE address [ ]?403
Pref CUG  []?
DTE address [ ]?404
Pref CUG  []?
DTE address [ ]?
```

```
XTP Config>add peer-router
Router's IP address?128.185.100.1

XTP config>add remote-dte
DTE address [ ]?201
Peer router's internal IP Address [0.0.0.0]?128.185.100.1
Peer router's internal IP Address [0.0.0.0]?
XTP config>exit

Config>
```

# Chapter 38. Configuring and Monitoring XTP

This chapter describe the XTP configuring and monitoring commands. It includes the following sections:

- "XTP Configuring Commands"
- "XTP Monitoring Commands" on page 451

## XTP Configuring Commands

This section describes the XTP configuring commands.

To access the XTP configuring environment, enter the **protocol xtp** command at the `Config>` prompt.

```
Config> p xtp
XTP config>
```

Enter the XTP configuring commands at the `XTP config>` prompt.

*Table 59. XTP Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Add | Adds an interface, peer router, closed user groups, remote DTE or PVC definitions. |
| Change | Changes a peer router, remote DTE or PVC definition. |
| Delete | Deletes a local DTE, peer router, closed user groups, remote DTE or PVC definition. |
| Enable-XTP | Activates the XTP forwarder. |
| Disable-XTP | Deactivates the XTP forwarder. |
| Set | Sets the value of the XTP Keepalive Timer. |
| List | Lists interfaces, peer routers, remote DTEs and PVC definitions. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Add

Adds a local X.25 node, a peer router, a remote X.25 node with corresponding routers, or a PVC from a local X.25 node to a remote X.25 node.

Wild card addressing is included in the XTP forwarder. When the local or remote DTE addresses are entered, they can contain a wild card character ( ? or * ). For additional information on the use of wildcards, see "DTE Address Wildcards" on page 433 .

**Syntax:**

**add** bi-cug
cug
local-dte
peer-router

remote-dte

pvc

**cug** Specifies the closed user group numbers for the XTP protocol. The first CUG you are prompted for is the preferred cug.

**Valid values:** 0 to 9999

**Default value:** None

**Example:**

```
add cug
Pref CUG []? 114
CUG (2) []? 314
CUG (3) []? 478
CUG (4) []?
```

**bi-cug** Specifies the bilateral closed user group numbers for the XTP protocol. The first bi-cug you are prompted for is the preferred bi-cug.

**Valid values:** 0 to 9999

**Default value:** None

**Example:**

```
add bi-cug
Pref BI-CUG []? 50
BI-CUG (2) []? 51
BI-CUG (3) []? 52
BI-CUG (4) []? 53
BI-CUG (5) []? 54
```

**local-dte**

Adds the X.25 DTE addresses, or the X.25 nodes, that communicate with the router on the specified interface.

You can configure multiple local nodes. However, only one local node can be configured if the option to allow inbound calls without a calling dte address has been selected.

**Example:**

```
add local-dte

Interface number [0]?4
Allow inbound calls without calling DTE address? (Y or N) [N]? y
DTE address [ ]?101
Pref CUG  []? 23
CUG (2)  []? 24
CUG (3)  []? 25
CUG (4)  []? 26
CUG (5)  []? 27
Pref BI-CUG []? 6
BI-CUG (2) []? 7
BI-CUG (3) []? 8
BI-CUG (4) []? 9
BI-CUG (5) []? 10
DTE address [ ]?
```

**peer-router**

Adds peer routers. Enter the internal IP addresses of the routers to which the remote X.25 nodes are connected. You can use these IP addresses to open TCP connections and transport X.25 packets that contain connection requests and X.25 data.

If the internal IP address you configure for the peer-router is this router's internal IP address, the software establishes a local XTP connection.

**Example:**

```
add peer-router

Router's internal IP Address [0.0.0.0]?128.185.100.2
Connection setup timeout [230]?
```

**remote-dte**

Adds remote X.25 nodes and corresponding routers. You can connect remote nodes with local X.25 nodes so they can exchange data. You must configure an IP address for each remote X.25 node you configure. Any request or data sent to this remote node goes to the router. The router then uses one of its local X.25 interfaces to forward the data to the X.25 node.

Define a remote DTE if this router is to initiate XTP connections to the remote DTE due to incoming calls from its local DTEs, or if the remote DTE is part of an XTP PVC definition.

To use Local XTP, the peer router address must be the internal address of the local router and that DTE address must be previously defined using the **add local** command.

**Example:**

```
add remote-dte

DTE address [ ]?301
Peer router's internal IP Address [0.0.0.0]?128.185.100.2
Peer router's internal IP Address [0.0.0.0]?
```

**pvc** Adds a PVC from a local X.25 node to a remote X.25 node.

Three things need to exist in order to activate a PVC configuration:

- An X.25 PVC from the router to the local X.25 node
- An X.25 PVC from the peer router to the remote X.25 node
- A TCP connection to the peer router where the remote node is resident

**Example:**

```
add pvc

Local PVC Number [1]?1
Local X.25 DTE address [ ]?100
Remote PVC Number [1]?1
Remote X.25 DTE address [ ]?301
```

**Notes:**

1. When you add PVCs to the router configuration, you also must configure the PVC in X.25. For details on configuring X.25 interfaces, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*
2. For Local XTP, you must define the PVC in both directions. You need this definition because the router is performing both local and remote functions. For example, to define Local PVC 8 and Remote PVC 10 when you are using Local XTP, you would do the following:

   ```
   add pvc

   Local PVC Number [1]?8
   Local X.25 DTE address [ ]?108
   Remote PVC Number [1]?10
   Remote X.25 DTE address [ ]?310
   add pvc

   Local PVC Number [1]?10
   Local X.25 DTE address [ ]?310
   Remote PVC Number [1]?8
   Remote X.25 DTE address [ ]?108
   ```

**Note:** When you add PVCs to the router configuration, you also must configure the PVC in X.25. For details on configuring X.25 interfaces, refer to *Software User's Guide for Nways Multiprotocol Access Services Version 3 Release 1*

## Change

Changes a peer router, remote DTE, or PVC from the XTP configuration.

**Syntax:**

**change** peer-router
remote-dte
pvc

**peer-router**
Changes specific peer routers from the XTP configuration.

**Example:**

```
change peer-router

Router IP Address [0.0.0.0]?128.185.100.2
```

**remote-dte**
Changes specific remote DTEs in the XTP configuration.

**Example:**

```
change remote-dte

DTE address [ ]?401
Peer router's internal IP Address [0.0.0.0]?128.185.100.2
Peer router's internal IP Address [0.0.0.0]?
```

**pvc** Changes PVC definitions from the XTP configuration.

**Example:**

```
change pvc

Local PVC number [1]?1
Local DTE address [ ]?301
```

## Delete

Deletes a local DTE, peer router, remote DTE, or PVC from the XTP configuration.

**Syntax:**

**delete** bi-cug
cug
local-dte
peer-router
remote-dte
pvc

**bi-cug** Deletes a bilateral closed user group number used by this interface.

**Valid values:**

**Y** Deletes the current CUG.

**N** Does not delete the current CUG.

**ALL** Deletes all remaining CUGs.

**Q** Stops deleting any remaining CUGs.

**Example:**

```
delete bi-cug
Delete Pref BI-CUG [Y]?
Delete BI-CUG (2) [Y]? N
Delete BI-CUG (3) [Y]? q
```

**cug** Deletes the closed user group numbers used by this interface. This command works similar to the **delete bi-cug** command.

**Example:**

```
del cug

Delete Pref CUG [Y]?
Delete CUG (2) [Y]?
Delete CUG (3) [Y]? q
```

**local-dte**
Deletes specific local interfaces from the XTP configuration.

**Example:**

```
delete local-dte

Interface number [0]?1
DTE address [ ]?101
Record deleted
```

**peer-router**
Deletes specific peer routers from the XTP configuration.

**Example:**

```
delete peer-router

Router IP Address [0.0.0.0]?128.185.100.2
Record deleted
```

**remote-dte**
Deletes specific remote DTEs from the XTP configuration.

**Example: `delete remote-dte`**

DTE address [ ]?**401**

**pvc** Deletes PVC definitions from the XTP configuration.

**Example:**

```
delete pvc

Local PVC number [1]?1
Local DTE address [ ]?301
Record deleted
```

## Enable

Activates the XTP forwarder.

**Syntax:** enable-xtp

**Example: `enable-xtp`**

## Disable

Deactivates the XTP forwarder.

**Syntax:** disable-xtp

**Example: `disable-xtp`**

## Set

Sets the XTP Keepalive Timer.

**Syntax:** keep-alive-timer

**Example:**

```
set keep-alive-timer

Keepalive timer in seconds [10]?60
```

## List

Lists the interfaces, peer routers, remote DTEs, or PVCs.

**Syntax:**

**list** all
cugs
keep-alive-timer
local-dtes
peer-routers
remote-dtes
pvcs
xtp-status

**all** Displays all the interfaces, peer routers, remote DTEs, and PVCs configured for XTP.

**Example:**

```
list all

STATUS: XTP-DISABLED

Local DTEs:

Interface      DTE Address
    1          44444                    Calling DTE address is optional
             Pref CUG     : 7777  Others : 9999 0
             Pref BI-CUG  : 0     Others :

    4          33333                    Calling DTE address is optional
             Pref CUG     : 1     Others : 2 3 4 5
             Pref BI-CUG  : 6     Others : 7 8 9 10


   Peer Routers    Connection Timeout

Remote DTEs:
      DTE Address    Peer Router(s)

PVCs:
Local PVC         Local DTE      Remote PVC        Remote DTE
 Number             Address        Number            Address

Pref CUG     : 114   Others : 314 478
Pref BI-CUG  : 1     Others : 1 1 1 1111


KEEP-ALIVE-TIMER: 10 seconds
```

**cugs** Lists the CUG and BI-CUG numbers defined for the XTP protocol.

**keep-alive-timer**
Displays all the Keepalive time configured for XTP.

**local-dtes**
Displays all the local DTEs configured for XTP.

**Example:**

```
list local-dtes

Local DTEs:
Interface        DTE Addr
   1             101 Calling DTE address is required
   2             201 Calling DTE address is required
```

**peer-routers**
Displays all the peer routers configured for XTP.

**Example:**

```
list peer-routers

Peer Routers:
128.185.100.2
128.185.100.3
```

**pvcs** Displays all the PVCs configured for XTP.

**Example-**

```
list pvcs

PVCs:

Local PVC      Local DTE      Remote PVC      Remote DTE
 Number          Address        Number          Address
  1                100              1              301
```

**remote-dtes**
Displays all the remote DTEs configured for XTP.

**Example:**

```
list remote-dtes

Remote DTEs:
DTE Address      Peer Router
    301         128.185.100.2
    401         128.185.100.3
    402         128.185.100.3
    403         128.185.100.3
    404         128.185.100.3
```

**xtp-status**
Displays the status of XTP indicating whether it is enabled or disabled.

**Example:**

```
list xtp-status

STATUS:  XTP-ENABLED
```

## XTP Monitoring Commands

This section describes the XTP monitoring commands. These commands allow you to display the current active interfaces, peer routers, remote DTE, PVCs and SVCs. They also allow you to dynamically add or delete interfaces, DTEs, or peer routers.

To display the XTP> prompt, enter **protocol xtp** at the monitoring (+) prompt:

```
+protocol xtp
X.25 Transport Console
XTP>
```

### XTP Monitoring Commands (Talk 5)

Enter the XTP monitoring commands at the XTP> prompt.

*Table 60. XTP Monitoring Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add | Dynamically adds local DTEs, remote DTEs, or peer routers |
| Delete | Dynamically deletes configurations for local DTEs, remote DTEs, or peer routers |
| List | Displays individual PVC or SVC statistics and general information |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Add

Adds an interface, peer router, or remote DTE to the XTP configuration.

**Syntax:**

**add** local-dtes
peer-router
remote-dtes

**local-dtes**

Adds a local interface to the XTP configuration.

**Example:**

```
add local-dtes

Interface number [0]?1
DTE address [ ]?101
```

**peer-router**

Adds a peer router to the XTP configuration.

**Example:**

```
add peer-router

Router's IP Address [0.0.0.0]?128.185.100.2
```

**remote-dtes**

Adds a remote DTE to the XTP configuration.

**Example:**

```
add remote-dtes

Peer router's IP Address [0.0.0.0]?128.185.100.2
DTE address [ ]?301
DTE address [ ]?
```

## Delete

Deletes a local DTE, peer router, or remote DTE from the router configuration.

**Syntax:**

**delete** local-dtes
peer-router
remote-dtes

**local-dtes**
Deletes a local interface from the XTP configuration.

**Example:**

```
delete local-dtes

Interface Number [0]?1
DTE address [ ]?101
DTE address [ ]?
```

**peer-router**
Deletes a peer router from the XTP configuration.

**Example: `delete peer-router`**

```
Router's IP Address [0.0.0.0]?123.185.100.2
```

**remote-dtes**
Deletes a remote DTE from the XTP configuration.

**Example:**

```
delete remote-dtes

DTE address [ ]?401
DTE address [ ]?
```

## List

Displays the current active interfaces, peer routers, remote DTEs, PVCs, and SVCs.

**Syntax:**

**list** all
xtp-status
local-dtes
peer-routers
remote-dtes
pvcs
pvc-detailed
pvcs-all-detailed
svcs
svc-detailed
svc-all-detailed

**all** Displays output of all list command options.

**example:**

```
list all


STATUS:  XTP-ENABLED
KEEP-ALIVE TIMER = 20 seconds


LIST OF LOCAL DTES
------------------
Interface     Local
  No           DTE
  1            101     Calling DTE address is required
  2            201     Calling DTE address is required


LIST OF PEER ROUTERS
--------------------
```

```
 Router             CNN      Number        Received           Sent
                   State    of  Ckts     Pkts    Bytes     Pkts    Bytes
128.185.100.3     Active      15          60      1533      12      142
128.185.100.2     Active      12          63      1620      10      130


LIST OF REMOTE DTES
-------------------
      Remote         Router
       DTE             IP
       404           128.185.100.3
       403           128.185.100.3
       402           128.185.100.3
       401           128.185.100.3
       301           128.185.100.2


LIST OF PVCS
------------
Index     Int    PVC    Local       Local   Remote      Remote
 No        No   State    LCN         DTE      LCN         DTE
  1         1   Active               100                 301


LIST OF SVCS   (list svcs)
------------
Index   Int   Logical    SVC        Local     Remote         Peer
  No     No   Channel   State        DTE        DTE        Router
  1       2       5      ACT     333333333333 4444444444444 3.3.3.3


SVC  1 IN DETAIL   (list svc-detailed)
----------------
Int    Log     SVC        Received            Sent             Dropped
 No    Chn    State      Pkts   Bytes      Pkts   Bytes      Pkts   Bytes

  2      5     ACT          2     116         2     106         0       0


LIST OF SVCS   (svcs-all-detailed)
------------
Int    Log     SVC        Received            Sent             Dropped
 No    Chn    State      Pkts   Bytes      Pkts   Bytes      Pkts   Bytes

  2      5     ACT          1       7         1       2         0       0
```

### xtp-status

Displays whether XTP is enabled/disabled, and the time specified for the Keepalive Timer.

#### Example:

```
list xtp-status

STATUS:  XTP-ENABLED
         KEEP-ALIVE-TIMER = 20 seconds
```

### local-dtes

Displays all the interfaces configured for XTP.

#### Example:

```
list local-dtes

LIST OF LOCAL DTES
------------------
Interface     Local
  No           DTE
  1            101     Calling DTE address is required
  2            201     Calling DTE address is required
```

### peer-routers

Displays all the peer routers configured for XTP.

#### Example:

```
list peer-routers

LIST OF PEER ROUTERS
--------------------
 Router             CNN      Number        Received           Sent
                   State    of  Ckts     Pkts    Bytes     Pkts    Bytes
128.185.100.3     Active      15          60      1533      12      142
128.185.100.2     Active      12          63      1620      10      130
```

**remote-dtes**

Displays all the remote interfaces configured for XTP.

**Example:**

```
 list remote-dtes

LIST OF REMOTE DTES
-------------------
      Remote         Router
       DTE             IP
       404           128.185.100.3
       403           128.185.100.3
       402           128.185.100.3
       401           128.185.100.3
       301           128.185.100.2
```

**pvcs** Displays all the PVCs configured for XTP.

**Example:**

```
list pvcs

LIST OF PVCS
------------
Index     Int    PVC   Local     Local    Remote      Remote
 No        No   State   LCN       DET      LCN         DTE
  1         1   Active            100                  301
```

**pvc-detailed**

Displays detailed information for a specific PVC definition. For a listing of Index numbers, enter **list all** at the xtp> prompt.

**Example:**

```
list pvc-detailed

PVC Index Number [1]?1

PVC  1 IN DETAIL
----------------
Int     PVC         Received            Sent             Dropped
 No    State      Pkts     Bytes    Pkts     Bytes    Pkts      Bytes
 1     ACTIVE      55       3220     35       2350     15        1870
```

**pvcs-all-detailed**

Displays detailed information for all PVC definitions.

**Example:**

```
list pvcs-all-detailed

LIST OF PVCS
------------
INT Local     PVC         Received            Sent             Dropped
 No  LCN     State      Pkts     Bytes    Pkts     Bytes   Pkts     Bytes
 1           ACTIVE      55       3220     35       2350    15       1870
```

**svcs** Displays all the SVCs definitions.

**Example:**

```
list svcs

LIST OF SVCS
------------
Index     Int LOG    SVC       Local      Remote  Peer
 No        No Chan  State       DTE         DTE   Router
  1         1       Active      200         401   3.3.3.3
  2         1       Active      200         402   3.3.3.3
  3         2       Active      200         403   3.3.3.3
  4         2       Active      200         404   3.3.3.3
```

**svc-detailed**

Displays information for specific SVC definitions.

**Example:**

```
list svc-detailed

SVC Index Number [1]?1
SVC  1 IN DETAIL
----------------
```

```
Int  LOG    SVC        Received            Sent            Dropped
 No  Chan  State     Pkts      Bytes    Pkts      Bytes   Pkts       Bytes
  1        ACTIVE     75       4220     55        3350    20         870
```

### svcs-all-detailed

Displays information for all the SVC definitions.

#### Example:

```
list svcs-all-detailed

LIST OF SVCS
------------
Index     Int Log    SVC         Received          Sent           Dropped
No        No  Chn  State      Pkts    Bytes   Pkts     Bytes   Pkts    Bytes
1          1       ACTIVE     4220     55     550        20      870
2          1       ACTIVE     3220     40     2350        15      970
3          2       ACTIVE     4003     50     3892        20      870
4          2       ACTIVE     3967     58     4167        12      800
```

# Chapter 39. Using Frame Relay Interfaces

This chapter describes how to use the Frame Relay interface 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) or a Forward Explicit Congestion Notification (FECN), the router initiates a controlled slowdown of traffic, thereby avoiding a complete FR network shutdown.

  The router can also initiate a slowdown of traffic when it receives a Consolidated Link Layer Management (CLLM) congestion message. CLLM is an optional part of the Frame Relay standards that provides additional management information about the operation of the frame relay network to attaching DTEs.
- *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.
- *Data compression* that supports the FRF.9 standard. See "Chapter 66. Using the Data Compression Subsystem" on page 801 for details.
- *Data encryption* using a proprietary encryption scheme. See "Chapter 70. Overview of Encryption" on page 843 for details.

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 39, 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.

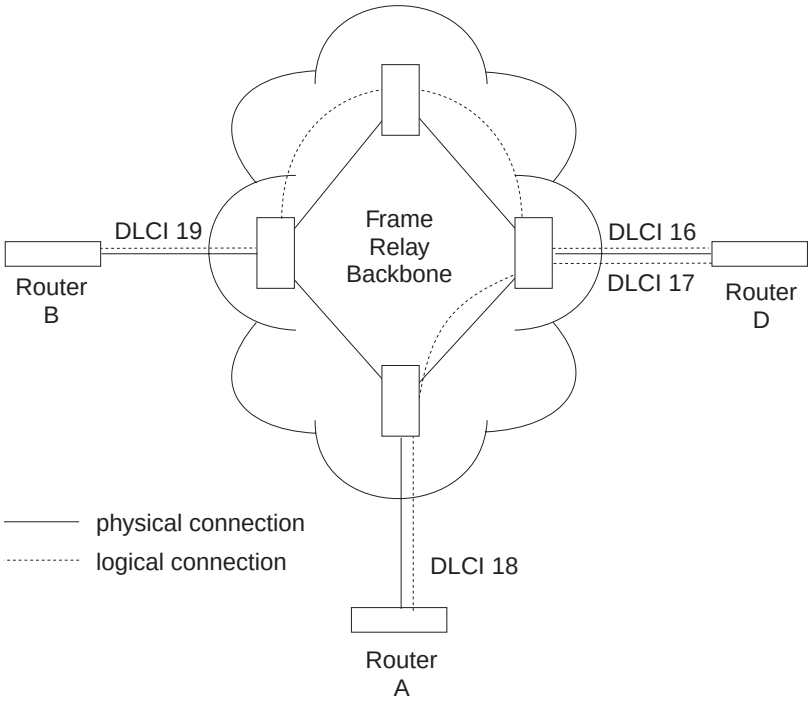



*Figure 39. DLCIs in Frame Relay Network*

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 does not 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 460).

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 40 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.

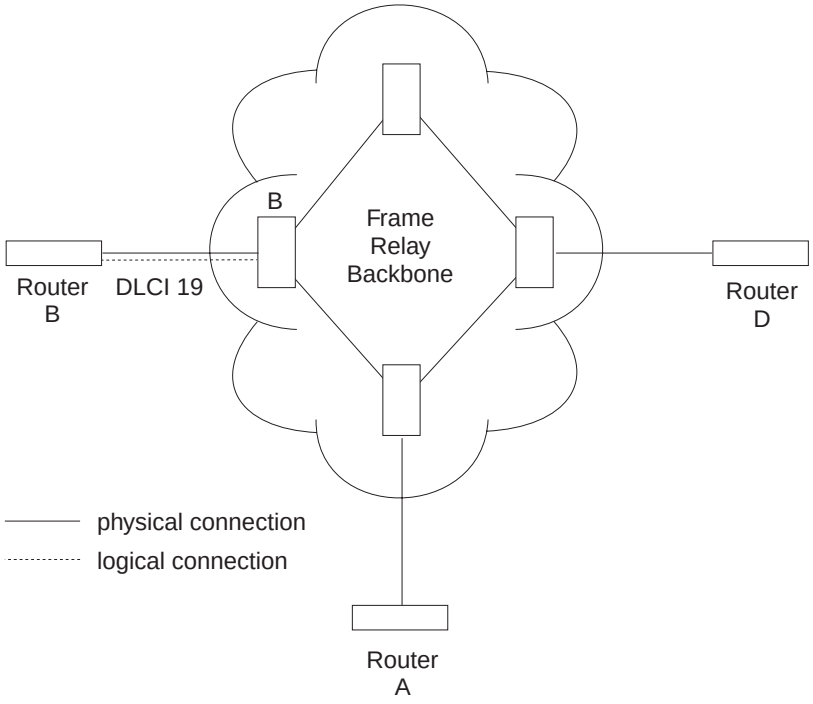



*Figure 40. DLCIs in Frame Relay Network*

When the Local Management 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 41 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.

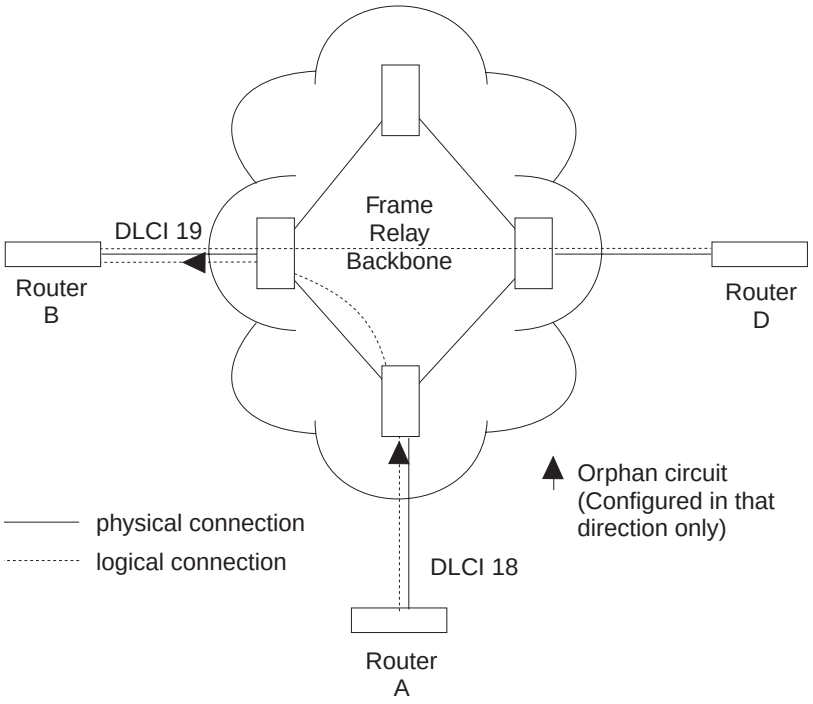



*Figure 41. Orphan Circuit*

## 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 42 illustrates a Frame-Relay frame format.

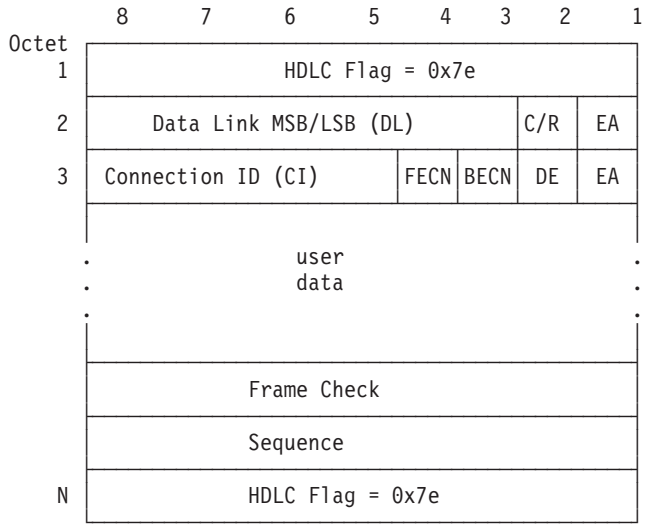



*Figure 42. Frame-Relay Frame Format*

### 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 to 8 of octet 2 and bits 5 to 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. You can configure the device to slow down data transmission in the direction from which it receives a FECN using the **enable throttle-transmit-on-fecn** command. You can also set the BECN bit in data frames sent to the originator of the FECN using the **enable notify-fecn-source** command.

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 algorithm 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 Frame Relay network may discard transmitted data exceeding CIR on a PVC. The DE bit can be set by the router to indicate that some traffic should be considered discard eligible. If appropriate, the Frame Relay network will discard frames marked as discard eligible which may allow frames that are not marked discard eligible to make it through the network. To identify traffic that is discard eligible:

1. Configure BRS on the Frame Relay interface and any FR circuits that has traffic that you are making discard eligible.
2. Assign a protocol or filter to a BRS traffic class using the **assign** command. You specify whether the DE bit should be set on for this protocol or filter traffic.

### 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.

**Note:** You can configure the use of a 32-bit CRC for a Frame Relay interface on a HSSI adapter.

## 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 61.

**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 61. 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.

For protocols that support ARP/InARP over Frame Relay, Frame Relay will only multicast a protocols packets over a circuit if a protocol address was either learned or configured for that circuit.

## 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 465.
- 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 465.)

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.

## Consolidated Link Layer Management (CLLM)

CLLM is an optional FR management function that is not widely supported by the industry but it has been adopted by some Frame Relay switch manufacturers. CLLM provides some of the same management information provided by LMI, in particular, outage notification. CLLM's main use is to provide asynchronous congestion notification to attaching devices. A single CLLM message may indicate outage or congestion for multiple PVCs. The Frame Relay protocol supports the following standards for CLLM: ANSI T1.618, ITU-T (CCITT) Q.922 Annex A, and ITU-T (CCITT) X.36 Annex C.

# 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 2 Mbps * reserved for the PVC. 64 Kbps or a single DS0 channel is most common.

***Note:** The maximum CIR value for a FR interface on a HSSI adapter is 52 Mbps.

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. You can also set the default CIR for all Frame Relay circuits on this interface using the **set CIR-defaults** 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 a CIR to orphan circuits based on the CIR defaults configured at the interface level. If you are relying on the orphan circuit to route important data and the CIR, Bc, and Be values from the network provider are different from the values configured at the interface level, it is recommended that you define a PVC instead of an orphan circuit. 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, Frame Relay uses a value of 1 second for Tc..

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. You can also set the default committed burst size for all Frame Relay circuits on this interface using the **set CIR-defaults** command.

The device assigns orphan circuits a committed burst size based on the default you set with the **set CIR-defaults** command. 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, Frame Relay used a value of 1 second for Tc.

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. Orphan circuits will receive a default excess burst size equal to the value set in the **set CIR-defaults** command. If you configure 0 for CIR, then you must configure a nonzero value for the excess burst (Be) size. You can also set the default excess burst size for all Frame Relay circuits on this interface using the **set CIR-defaults** command.

## 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, 64 Kbps is used as the line speed.

For dial circuits using Channelized T1/E1 as the base net, the line speed is 64 Kbps times the number of timeslots assigned or 56 Kbps if you set the bandwidth of the Channelized circuit to 56 Kbps. For example, if you set the number of timeslots for a Channelized circuit to 3, the line speed is 192 Kbps (3 * 64 Kbps).

If the dial circuit is mapped to a V.25bis base interface, the line speed of the V.25bis interface is used for the FR dial circuit.

## 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 1500 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 nonzero, the maximum information rate is calculated using CIR, Bc, and Be as follows:

```
( Bc + Be )
```

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 64 000 and the MIR adjustment percentage is set to 25%, then the initial VIR would be equal to 16 000 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 available throughput allows.

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 step-wise 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.

**Note:** Even with compression enabled, the device uses the uncompressed size of frames to determine if the VIR is being exceeded.

## 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 a frame to notify the DTEs at each end of a PVC that congestion is occurring. FECN indicates that congestion is occurring in the same direction from which the frame was received; the sender is causing the congestion. BECN indicates that the frames sent by this DTE are causing network congestion.

Optionally, the network can use CLLM messages to convey congestion information. CLLM messages are sent only to the congestion source and should be treated similarly to BECN messages by the DTE.

The example in Figure 43 on page 470 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. FR switches often set BECN in multiple frames after reaching a congestion threshold. In order for FR to avoid overreacting to network congestion when the network is setting multiple frames with BECN, FR will decrease a PVC's VIR at most once every second. This allows the VIR to decrease gradually. As the router receives consecutive frames without BECN, the VIR gradually rises to the maximum IR.

Depending on the operation of the FR network, it may be necessary for the device to throttle down the PVC's VIR when the device receives a FECN to minimize the overall amount of traffic being offered to the network as quickly as possible. Reducing the overall load on the network reduces the number of packets discarded

for all PVCs to relieve congestion. Enabling the ***throttle-transmit-on-fecn*** parameter, along with either the CIR or congestion monitoring options, causes the device to treat a FECN like a BECN thus reducing overall FR network congestion when any congestion notification is received. Use the throttle-transmit-on-fecn parameter only in FR networks whose queuing methods do not provide dedicated buffers for both input and output. If the ***throttle-transmit-on-fecn*** is enabled, FR will decrease a PVC's VIR at most once every second for each BECN or FECN received.

Some FR network switches set FECN to indicate congestion but do not set BECN. To provide congestion notification to the source of the congestion, enable the ***notify-fecn-source*** parameter allowing the device to set BECN in frames that it transmits over a PVC on which it has received a FECN. This action provides a signal to the device that is causing the network congestion to throttle down its PVC's VIR.

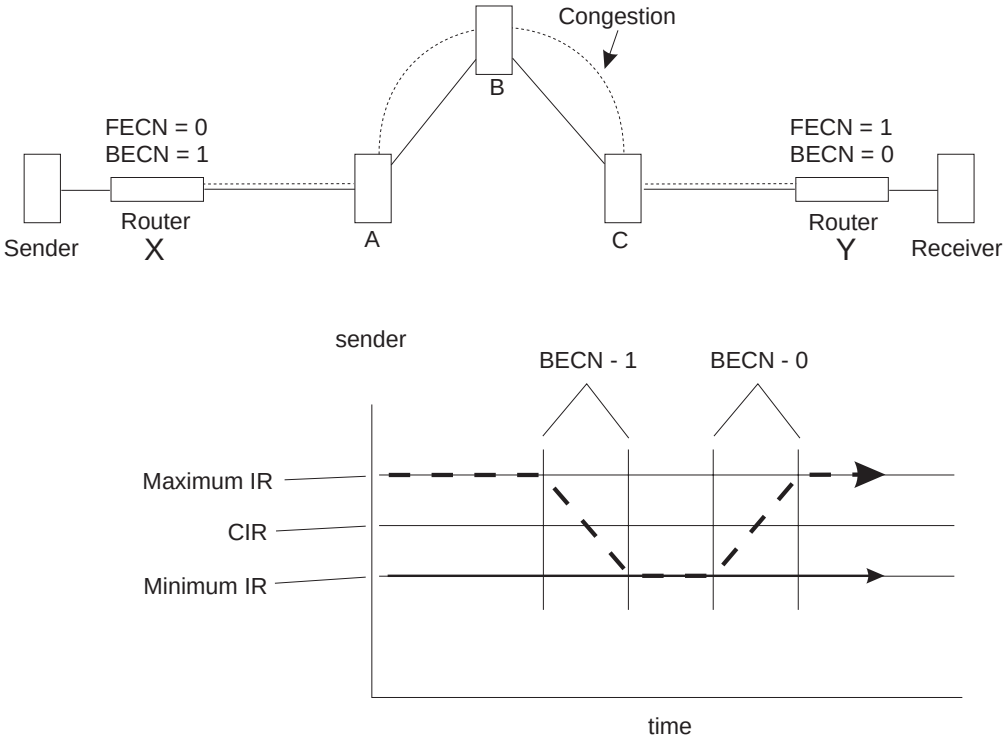



*Figure 43. Congestion Notification and Throttle Down*

**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.

Similarly, if the network provider supports CLLM, you can configure Frame Relay to *throttle down* its transmit rate for PVCs contained in a CLLM message. CLLM messages contain a cause code that indicates the type and severity of the problem being reported. The device reacts differently depending on the cause code and the CIR configured for each PVC contained in the CLLM message. When the device receives a CLLM message that indicates:

- A short-term condition, and the configured CIR for the PVC is nonzero, the Frame Relay protocol will throttle the transmit rate for the affected PVCs by the configured IR decrement percentage.
- A long-term condition, the Frame Relay protocol will set the transmit rate for the affected PVCs to the calculated minimum information rate.
- Facility or equipment failure or maintenance action, or if the CIR was configured as zero, the FR protocol will continue to transmit any queued data for the affected PVCs but will not accept any more outgoing packets from the upper layer protocols until the congestion condition is cleared.

Once a CLLM message for a PVC has been received, if the device does not receive any CLLM messages or BECNs within the *Ty* timer period or if a frame without a BECN is received, the device will consider the congestion condition cleared and gradually return the PVC to its configured transmission rates. If you are using CLLM to control congestion, you must not configure DLCI 1007 for any other use.

## Bandwidth Reservation over Frame Relay

For information on bandwidth reservation over Frame Relay, refer to “Chapter 57. Using Bandwidth Reservation and Priority Queuing” on page 681 through “Chapter 58. Configuring and Monitoring Bandwidth Reservation” on page 699.

## 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 2 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 57. Using Bandwidth Reservation and Priority Queuing” on page 681.
- **Configure Discard Eligibility.** You can configure Discard Eligibility (DE) congestion control using Bandwidth Reservation. For information on configuring Discard Eligibility, refer to “Chapter 57. Using Bandwidth Reservation and Priority Queuing” on page 681.
- **Configure Data Compression.** You can configure data compression for Frame Relay. For information on configuring data compression, refer to “Chapter 66. Using the Data Compression Subsystem” on page 801.

## 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 62 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 62. 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
```

# Chapter 40. Configuring and Monitoring Frame Relay Interfaces

This chapter describes the Frame Relay configuration and operational commands and includes the following sections:



**Note:** For information on monitoring bandwidth reservation over Frame Relay, refer to "Chapter 58. Configuring and Monitoring Bandwidth Reservation" on page 699 .

## Frame Relay Configuration Commands

This section describes 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 63. Frame Relay Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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 | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

**Note:** In this section, the terms *circuit number* and *PVC* are synonymous with the term DLCI (Data Link Circuit Identifier).

# Add

Use the **add** command to add a PVC, Required PVC group, or destination protocol address supported by the Frame Relay interface.

**Syntax:**

**add** permanent-virtual-circuit . . .

protocol-address . . .

pvc-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 []?
Do you want to have data compression performed [Y]?
Do you want to have data encryption performed [N]? y

Data encryption requires a key that is 16 hexadecimal characters long
You will be asked to enter the key twice for security reasons

Please enter the key for the first time now

A valid encryption key has been entered

Please confirm the key by entering it again

The encryption keys match - the key has been accepted
```

**Circuit Number**

Indicates the circuit number for this PVC.

**Valid Values:** 16 to 1007.

**Note:** If you are configuring CLLM to help control congestion, you cannot configure 1007 as a PVC.

**Committed Information Rate**

Indicates the committed information rate (CIR). The CIR can be either 0 or a value in the range 300 bps to 2 048 000 bps. For more information, see “Committed Information Rate (CIR)” on page 465. The maximum is the value of the default CIR configured for the interface.

**Note:** The default for a FR interface on a HSSI adapter is 52 000 000 bps, not 2 048 000 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 maximum value is value of the default committed burst configured for the interface.

**Notes:**

1. The default for a FR interface on a HSSI adapter is 52 000 000 bps, not 2 048 000 bps.
2. 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 466.

**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 2 048 000 bits. The maximum value is the value configured for excess burst size for the interface. For additional information, see "Excess Burst (Be) Size" on page 466.

**Note:** The default for a FR interface on a HSSI adapter is 52 000 000 bps, not 2 048 000 bps.

**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 displayed only 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**

Enables you to specify 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.

**Do you want to have compression performed**

Enables you to specify whether or not the circuit will compress data packets. This question appears only if compression is enabled on the interface.

**Note:** If you enable compression on a PVC and exceed the interface's compression PVC limit, you will get a message. Compression will be performed on the circuit, if possible – that is, the active compression limit has not been exceeded when the circuit becomes active.

**Do you want to have data encryption performed**

Enables you to specify whether or not the circuit will encrypt data packets. This question appears only if encryption is enabled on the interface. The prompts for the encryption key will only appear if you respond "yes" (or "y") to this question.

**Specifying the Encryption Key:** The encryption key is 16 hexadecimal characters long. You must specify the encryption key as a value between X'0000000000000000' and X'FFFFFFFFFFFFFFFF'.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88 .

**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 relearned 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** *groupname*

Adds a Required PVC group name.

## 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 []?
Do you want to have data compression performed []?
Do you want to have data encryption performed []?
```

**Circuit Number**
Indicates the circuit number for this PVC.

**Valid Values:** 16 to 1007.

**Note:** If you are configuring CLLM to help control congestion, you cannot configure 1007 as a PVC.

**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 for an interface is 64000 bps, but the default for an individual circuit is the value configured with the **set cir-defaults** command.

**Note:** The default for a FR interface on a HSSI adapter is 52 000 000 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 2 048 000 bits. The default for an interface is 64000 bits, but the default for an individual circuit is the value configured with the **set cir-defaults** command.

**Note:** The default for a FR interface on a HSSI adapter is 52 000 000 bps.

**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 2 048 000 bits. The default for an interface is 64000 bps, but the default for an individual circuit is the value configured with the **set cir-defaults** command.

**Note:** The default for a FR interface on a HSSI adapter is 52 000 000 bps.

**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**
Enables you to specify 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.

**Do you want to have data compression performed**
Enables you to specifies whether or not the circuit will compress data packets. This question appears only if compression is enabled on the interface.

**Note:** If you enable compression on a PVC and exceed the interface's compression PVC limit, you will get a message. Compression will be performed on the circuit, if possible – that is, the active compression limit has not been exceeded when the circuit becomes active.

**Do you want to have data encryption performed**
Enables you to specify whether or not the circuit will encrypt data packets. This question appears only if encryption is enabled on the interface.

The default for the question depends on the current encryption state on the PVC. If the PVC is not currently encrypting data and you change the state to encrypt data, the software prompts you for the encryption key as

described in the **add permanent-virtual-circuit** command. See 477 for details about entering the encryption key.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

## Disable

Use the **disable** command to disable those features previously enabled using the **enable** command.

**Syntax:**

**disable**
- cir-monitor
- cllm
- compression
- congestion-monitor
- dn-length-field
- encryption
- lmi
- lower-dtr
- multicast-emulation
- no-pvc
- notify-fecn-source
- orphan-circuits
- protocol-broadcast
- throttle-transmit-on-fecn

**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 468 for more information.

**cllm** Disables the device from *throttling down* in response to a CLLM message. The default is disabled. See “Circuit Congestion” on page 468.)

**compression**
Disables compression on the interface. Compression will not be performed for any PVC.

**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 468 for more information. The default setting for this feature is enabled.

**dn-length-field**
Prevents inter-operation with implementations of DECnet Phase IV over Frame Relay that require a length field to precede DECnet packets in

Frame Relay frames, but allows inter-operation 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

**encryption**
Disables encryption on the interface. Even though the PVCs on this interface may be encryption capable, encryption will not take place.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**lmi**

**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.

**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**.

**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 463.

**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.

**notify-fecn-source**
Disables setting a BECN bit on the first packet destined to a device from which the router received a packet with the FECN bit set. See “Circuit Congestion” on page 468 for more information.

**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 that will be used on the interface.

**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 463. The default setting for this feature is enabled.

**throttle-transmit-on-fecn**
Prohibits the device from *throttling down* the transmission of packets in response to a packet with a FECN bit set on. The default is disabled. See “Circuit Congestion” on page 468 for more information.

## Enable

Use the **enable** command to enable Frame Relay features.

**Syntax:**

**enable**
- cir-monitor
- cllm
- compression
- congestion-monitor
- dn-length-field
- encryption
- lmi
- lower-dtr
- multicast-emulation
- notify-fecn-source
- no-pvc
- orphan-circuits
- protocol-broadcast
- throttle-transmit-on-fecn

**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 468 .

**Note:** To maximize throughput for circuits running data compression, you should not enable CIR monitoring on the same interface on which

you have enabled compression. Because the device uses the uncompressed size of frames to determine if the VIR of a PVC is being exceeded and compressed frames will require less bandwidth, the CIR of a PVC will be under-utilized if the device strictly monitors and does not exceed the configured CIR. Instead, congestion monitoring can be used to allow the device to react to congestion indications sent by the FR network to avoid frame loss.

**cllm** Enables the device to *throttle down* in response to a CLLM message. Contact your FR network provider to see whether this support is available. See "Circuit Congestion" on page 468 for more information.

**compression**
Enables compression on the interface. All compression-capable PVCs on the interface can compress data packets, provided that contexts are available and the active compression PVC limit has not been exceeded. (See "Chapter 66. Using the Data Compression Subsystem" on page 801 for details.)

**Note:** To maximize throughput for circuits running data compression, you should not enable CIR monitoring on the same interface on which you have enabled compression. Because the device uses the uncompressed size of frames to determine if the VIR of a PVC is being exceeded and compressed frames will require less bandwidth, the CIR of a PVC will be under-utilized if the device strictly monitors and does not exceed the configured CIR. Instead, congestion monitoring can be used to allow the device to react to congestion indications sent by the FR network to avoid frame loss.

**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 469.

**dn-length-field**
Supports inter-operation 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.

**encryption**
Enables encryption on the interface. All PVCs that are configured as encryption enabled, will encrypt all transmitted data.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**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 472. 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.

**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**.

**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 463. The default for this parameter is enabled.

**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.

**notify-fecn-source**

Enables setting a BECN bit on the first packet destined to a device from which the router received a packet with the FECN bit set. Use this parameter to enhance the congestion control mechanisms of the device in a network whether the FR switches do not themselves set BECN but set FECN. See “Circuit Congestion” on page 468 for more information.

**orphan-circuits**
Enables the use of all non-configured orphan circuits. The default for this feature is enabled. See “Orphan Circuit CIR” on page 466 for information about the default CIR values.

**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.

**throttle-transmit-on-fecn**
Enables the device to *throttle down* the transmission of packets in response to a packet with a FECN bit set on. Use this parameter to minimize overall FR network congestion whenever a congestion indication is received. It causes the device to react to a FECN in the same way that it reacts to a BECN.

## List

Use the **list** command to display currently configured management and PVC information.

**Syntax:**

**list** all
hdlc
lmi
permanent-virtual-circuits
protocol-address
pvc-groups

**all** Displays the Frame Relay configuration. The display is a combination of the **list hdlc,** the **list lmi**, and the **list permanent virtual circuits** commands.

See **list hdlc** and **list lmi** for descriptions of the parameters.

**hdlc** Displays the Frame Relay High-Level Data Link Control (HDLC) configuration.

**Example:**

```
list hdlc
                        Frame Relay HDLC Configuration


 Maximum frame size     = 2048
 Encoding               = NRZ
 Idle state             = Flag
 Clocking               = External
 Cable type             = V.35 DTE
 Line speed (bps)       = 64000
 Transmit delay         = 0
 Lower DTR              = Enabled
```

**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.

**Maximum frame size**
Indicates the maximum frame size 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.

**Notes:**

1. For a FR dial circuit interface, only the maximum frame size is displayed.
2. For FR interfaces on the HSSI adapter, the **list hdlc** command only displays a subset of the HDLC parameters shown above.

**lmi** Displays logical management and related configuration information about the Frame Relay interface.

**Example:**

```
list lmi
                        Frame Relay Configuration


  LMI enabled            =  Yes   LMI DLCI                      =    0
  LMI type               =  ANSI  LMI Orphans OK                =  Yes
  CLLM enabled           =   Yes  Timer Ty seconds    = 10

  Protocol broadcast     =   Yes  Congestion monitoring         =  Yes
 Emulate multicast       =   Yes  CIR monitoring                =   No
  Notify FECN Source     =   Yes  Throttle Transmit on FECN     =  Yes

  Data compression       =  Yes   Orphan compression           =   No
  Compression PVC limit  =   10   Number of compression PVCs    =    5 1
  Data encryption        =  Yes  Number of encryption circuits  =    1 2

  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
  Default CIR            = 64000  Default Burst Size             = 64000
  Default Excess Burst   =     0
```

**Notes:**

1. This line appears only when data compression is on (yes).
2. This line appears only when data encryption is on (yes).

**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.

**CLLM Enabled**
Indicates whether CLLM is enabled on the Frame Relay interface.

**Timer Ty seconds**
Indicates the amount of time that must elapse without the device receiving any CLLM messages or BECNs before the device considers a congestion condition cleared and gradually return the PVC to its configured transmission rate.

**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.

**Notify FECN Source**
Indicates whether this device sets a BECN bit on the first packet destined to a device from which the router received a packet with the FECN bit set.

**Throttle Transmit on FECN**
Indicates whether the device will *throttle down* the transmission of packets in response to a packet with a FECN bit set on.

**Data compression**
Indicates whether this interface has data compression enabled.

**Data encryption**
Indicates whether this interface has data encryption enabled and the number of circuits that are encryption capable.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88 .

**Orphan compression**
Indicates whether orphan circuits on this interface will have data compression enabled.

**Note:** Enabling compression on orphan circuits will decrease the number of available compression contexts available for the native PVCs on the device.

**Compression PVC limit**
Indicates the maximum number of PVCs that can participate in data compression.

**Number of compression PVCs**
Indicates the current number of PVCs compressing data.

**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.

**Default CIR**
The committed information rate, in bits per second, used as the default for PVCs on this interface.

**Default Burst Size**
The committed burst size, in bits, used as the default for PVCs on this interface.

**Default Excess Burst Size**
The excess burst size, in bits, used as the default for PVCs on this interface.

**permanent-virtual-circuits**
Displays all the configured PVCs on the Frame Relay interface.

**Example:**

```
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
@ = circuit is data compression capable
$ = circuit is data encryption capable
```

**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 223 for an explanation of each of these commands.

**Note:** The **LLC** command is supported only if APPN is in the software load.

**Syntax:**

**llc**

## Remove

Use the **remove** command to delete any PVC, Required PVC group, or protocol-address previously added using the **add** command.

**Syntax:**

**remove** permanent-virtual-circuit . . .
protocol-address
pvc-group

**permanent-virtual-circuit** *pvc#*

Deletes any configured PVC in the range 16 to 1007.

**Notes:**

1. When you delete a PVC that is running compression, the interface decreases the count of active compression PVCs. If this action brings the count of compression PVCs below the limit, you will receive a message to that effect.
2. When you delete a PVC that is running encryption, the interface decreases the count of active encryption PVCs.

   **Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88 .

**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]?
```

**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.

**pvc-group** *groupname*

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]?**

# 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:**

**set**
- cable*
- cir-defaults
- clocking*
- crc-type*
- encoding*
- frame-size
- idle . . .*
- ir-adjustment . . .
- line-speed*
- lmi-type n1-parameter
- n2-parameter
- n3-parameter
- p1-parameter
- t1-parameter
- transmit-delay . . .*
- ty-parameter

*** 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 |

| Physical Interface Link Type | Data Connection Type |
|---|---|
| V36 | DTE, DCE |
| X21 | DTE, DCE |
| HSSI | DTE, DCE (see note) |

**Note:** When a HSSI DCE cable is used, the other device must also be configured to use a HSSI DCE cable.

**cir-defaults**

Sets the default values for the circuit congestion parameters. The parameters are:

**cir** Sets the default value of *cir* to the value provided by a Frame Relay network provider.

**Valid Values:** 0 or 300 to 204 800 bps

**Default Value:** 64 000

For HSSI, the maximum value that can be configured is 52 000 000 bps.

**bc** Sets the default value of *bc* to the value provided by a Frame Relay network provider.

**Valid Values:** See "Committed Burst (Bc) Size" on page 466

**Default Value:** 64 000

For HSSI, the maximum value that can be configured is 52 000 000 bps.

**be** Sets the default value of *be* to the value provided by a Frame Relay network provider.

**Valid Values:** See "Excess Burst (Be) Size" on page 466

**Default Value:** 0

For HSSI, the maximum value that can be configured is 52 000 000 bps.

**Example:**

```
FR 6 config> set cir-default
Default Committed Information Rate (CIR) in bps [64000]? 48000
Default Committed Burst Size (Bc) in bits [64000]? 40000
Default Excess Burst Size (Be) in bits [0]? 52000
```

**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 64 on page 496.

**Note:** **Clocking** is set to *external* if the cable type is *HSSI DTE* and is set to *internal* if the cable type is *HSSI DTE* and is not configurable.

**crc-type [crc-ccitt-16 or crc-ccitt-32]**
CRC type can be configured as either 16-bit CRC or 32-bit CRC. The default is **crc-ccitt-16**.

**Note:** CRC type can only be configured for a FR interface on a HSSI adapter.

**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.

**Note:** Encoding is set to NRZ for a FR interface on a HSSI adapter and is not configurable.

**frame-size** *#*
Sets the maximum size of the network layer portion of the frames transmitted and received on the interface. This maximum size includes the 2-byte DLCI address and the user data shown in figure 39-4. The size you configure must be consistent with the maximum frame size supported by the Frame Relay switch and by the other FR DTEs in the Frame Relay network. Values are 262 to 8190. The default is 2048. Since the configured frame size includes the DLCI address and the FR RFC 1490 multi-protocol encapsulation header, the maximum protocol packet size that can be transmitted is less than the configured frame size and is protocol dependent. The following table shows how many bytes to subtract from the configured frame size to determine the maximum protocol packet size that can be transmitted and received on the interface.

| | |
|---|---|
| IP | 4 bytes |
| IPX | 10 bytes |
| Appletalk Phase 2 | 10 bytes |
| DECnet Phase IV (DNA IV) | 12 bytes |
| Banyan Vines | 10 bytes |
| OSI | 10 bytes |
| Bridging | 10 bytes |
| APPN | 58 bytes (see note) |

**Note:** Assumes worst case for APPN BAN where a T/R MAC address header and LLC header are added in addition to the FR header bytes.
If FR data encryption is enabled then you must subtract up to an additional 12 bytes.

**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.

**Note:** Idle is set to **flag** for a FR interface on a HSSI adapter and is not configurable.

**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.

*Table 64. 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 |
| 1-port HSSI | 22 368 000 bps or 44 736 000 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 65 to determine the line speeds supported for the serial adapters when external clocking is used.

*Table 65. 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 |
| 1-port HSSI | 1 544 000 bps to 52 000 000 bps |

**lmi-type [rev1 or ansi or ccitt]**

Sets the management type for the interface. See “Enabling Frame Relay Management” on page 472 for details on setting Frame Relay management. The default is type **ansi** enabled.

*Table 66. 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. |

**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.

**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.

**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.

**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.

**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.

**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 67 on page 498 lists the units and range values for serial interfaces.

**Note:** If you configure a non-zero transmit delay for a FR interface on the 8-port EIA-232E adapter, 6-port V.35/V.36 adapter, or 8-port X.21 adapter, you must configure the line speed using the **set line-speed** command.

*Table 67. Transmit Delay Units and Range for the 2216 Serial Interface*

| Unit | Minimum | Maximum |
|---|---|---|
| Extra Flags | 0 | 15 |

**ty-parameter** *time*

Configures the interval after which the device considers an existing congestion condition indicated by the receipt of a CLLM message to be cleared. If the device receives a CLLM message before the timer expires, the device resets this timer.

**Valid Values:** 5 to 30 seconds.

**Default Value:** 11 seconds.

## Accessing the Frame Relay Monitoring Prompt

To access the Frame Relay operating 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 Monitoring
FR 2 >
```

## Frame Relay Monitoring Commands

This section summarizes and then explains the Frame Relay Monitoring commands. Use these commands to gather information from the database. Table 68 shows the commands.

*Table 68. Frame Relay Monitoring Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 monitoring prompt. |
| Set | Sets CIR, Committed Burst Size, and Excess Burst Size for a Frame Relay PVC. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

**Note:** In this section, the terms *circuit number* and *PVC* are equivalent to the term *data link circuit identifier (DLCI)*.

## 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:**

**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:**

| | |
|---|---|
| **di**sable | cir-monitor |
| | cllm |
| | congestion-monitor |
| | notify-fecn-source |
| | throttle-transmit-on-fecn |

## 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:**

| | |
|---|---|
| **e**nable | cir-monitor |
| | cllm |
| | congestion-monitor |
| | notify-fecn-source |
| | throttle-transmit-on-fecn |

## List

Use the **list** command to display statistics specific to the data-link layer and the Frame Relay interface.

**Syntax:**

| | |
|---|---|
| **list** | all |
| | circuit . . . |
| | lmi |

permanent-virtual-circuits

pvc-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.

**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

 Compression capable  =       Yes  Operational         =    Yes
 R-Rs received        =         0  R-Rs transmitted    =      0
 R-As received        =         0  R-As transmitted    =      0
 R-R mode discards    =         0  Enlarged frames     =      0
 Decompress discards  =         0  Compression errors  =      0
 Compression ratio    = 1.72 to 1  Decompression ratio = 1.10 to 1

 Encryption capable   =        Yes  Operational         =          Yes
 Encryption errors    =          0  Decryption errors   =            0
 Rcv error discards   =          0

 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 non-configured 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.

**Compression capable**
Indicates whether the circuit can compress data packets.

**Operational**
Indicates whether compression is active on the circuit. When this is yes, data is being compressed on this link.

**R-Rs received**
Indicates the number of Reset-Request packets sent by the peer decompressor. A peer decompressor sends a Reset-Request whenever the peer detects that it is out of synch with its peer compressor. If this number increases rapidly, packets are being lost or corrupted on this circuit.

**R-Rs transmitted**
Indicates the number of Reset-Request packets sent since compression started on the circuit. If this number increases rapidly, packets are being lost or corrupted on this circuit.

**R-As received**
Indicates the number of Reset-Acknowledgements received in response to Reset-Requests. The compressor also sends out this packet to signal that it has reset its compression history.

**R-As transmitted**
This is the number of Reset-Acknowledgements sent to the peer.

**R-R mode discards**
Indicates the number of compressed data frames that were discarded while waiting for an R-A after sending out an R-R.

**Enlarged frames**
This is a count of the frames that could not be compressed. Usually an incompressible frame is sent in its uncompressed format within a special compression frame type allowing the compressor and decompressor to remain synchronized.

**Decompress discards**
Indicates the number of compressed frames that were discarded because of decompression errors.

**Compression errors**
Indicates the number of frames that had compression errors which were transmitted in an uncompressed form.

**Compression ratio**
Indicates the approximate effectiveness of the compressor.

**Decompression ratio**
Indicates the approximate effectiveness of the decompressor.

**Encryption capable**
Indicates whether this circuit is encryption enabled.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88 .

**Operational**
Indicates whether encryption is active on the circuit. When this is yes, data is being encrypted on this link.

**Encryption errors**
Indicates the number of frames that had encryption errors.

**Decryption errors**
Indicates the number of frames that had decryption errors.

**Rcv error discards**
Indicates the number of compressed frames that were discarded because of reception problems.

**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
  CLLM enabled  =   Yes  Timer Ty seconds =   11
  Last CLLM cause code = Network congestion - short term (0x02)
  Protocol broadcast   =     Yes  Congestion monitoring     =   Yes
  Emulate multicast    =     Yes  CIR monitoring            =    No
  Notify FECN source   =      No  Throttle transmit on FECN =    No
  PVCs P1 allowed      =      64  Interface down if no PVCs =    No
  Line speed (bps)     =   64000  Maximum Frame size        =  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
  Default CIR          =   65636  Default burst size        = 64000
  Default Excess Burst =       0

  Current receive sequence  =  0
  Current transmit sequence =  0
  Total status enquires     =  0    Total status responses    =    0
  Total sequence requests   =  0    Total responses           =    0

  Data compression enabled  =  Yes  Orphan compression              =   No
  Compression PVC limit     = None  Active compression PVCs   =    1

  Data encryption enabled   =  Yes  Active encryption circuits =    1

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 non-configured circuits learned from Frame Relay management are available for use (yes or no).

**CLLM enabled**
Specifies whether this circuit will throttle transmission on receiving CLLM frames.

**Timer Ty seconds**
Indicates the value of the CLLM Ty timer. This field is only displayed if CLLM is enabled.

**Last CLLM cause code**
Indicates the congestion cause code given in the last CLLM message received or ***None*** if no CLLM messages have been received. This field is only displayed if CLLM is enabled.

**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.

**Default CIR**
Specifies the default CIR for this interface.

**Default Burst Size**
Specifies the default burst size for this interface.

**Default Excess CIR**
Specifies the default excess burst size for this interface.

**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 responses**
Indicates the total number of sequence number responses that the Frame Relay interface has received from the Frame Relay management entity.

**Data compression enabled**
Indicates whether data compression is enabled on this interface.

**Orphan compression**
Indicates whether orphan circuits on this interface will have data compression enabled.

**Note:** Enabling compression on orphan circuits will decrease the number of available compression contexts available for the native PVCs on the device.

**Compression PVC limit**
Specifies the maximum number of PVCs that can compress data on this interface.

**Active compression PVCs**
Specifies the number of PVCs currently compressing data on this interface.

**Data encryption enabled**
Indicates whether data encryption is enabled on this interface.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88 .

**Active encryption circuits**
Indicates the number of PVCs that are currently encrypting data.

**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
     18    Boston                 No    &#P/A             0         0
     19    Orlando                No     *P/A             0         0
     20    Port Royal             No     $P/A             0         0
     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
    @ - Data compression capable but not operational
    & - Data compression capable and operational
    $ - Data encryption capable but not operational
    % - Data encryption capable and operational
```

**Circuit#**
Indicates the number of the PVC.

**Circuit Name**
Name of the circuit, an ASCII string.

**Orphan Circuit**
Indicates whether the PVC is a non-configured 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 227 for an explanation of each of these commands.

**Syntax:**

**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:**

**set** circuit . . .

ir-adjustment . . .

**circuit** *circuit# cirvol bcval beval*

Sets the values for Committed Information Rate (CIR), Committed Burst Rate, and Excess Burst Rate for the specified PVC.

**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 465.

**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 466.

**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 466.

**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]?
```

## Frame Relay Interfaces and the GWCON Interface Command

While Frame Relay interfaces have a monitoring 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 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands” on page 99)

## 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 monitoring 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
Nt Nt'  Interface                         Passed      Failed      Failed
3   2    FR/1                                    1             0              0

    Frame Relay MAC/data-link on ISDN Primary Rate interface
```

# Chapter 41. Using Point-to-Point Protocol Interfaces

This chapter describes how to use 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 513
- "The PPP Network Control Protocols" on page 521
- "PPP Authentication Protocols" on page 517

See "Chapter 43. Using the Multilink PPP Protocol" on page 561 and "Chapter 44. Configuring and Monitoring Multilink PPP Protocol (MP)" on page 565 for information about using the Multilink PPP Protocol.

## PPP Overview

PPP provides a method for transmitting protocol datagrams at the Data Link Layer over serial point-to-point links. PPP provides the following services:

- Link Control Protocol (LCP) to establish, configure, and test the link connection.
- Encapsulation protocol for encapsulating protocol datagrams over serial point-to-point links.
- Authentication protocols (APs) to validate the identity of a peer (remote) unit, and to submit your own identity to the peer for validation.
- Network Control Protocols (NCPs) for establishing and configuring different network layer protocols. PPP allows the use of multiple network layer protocols.

Figure 44 shows some examples of point-to-point serial links.

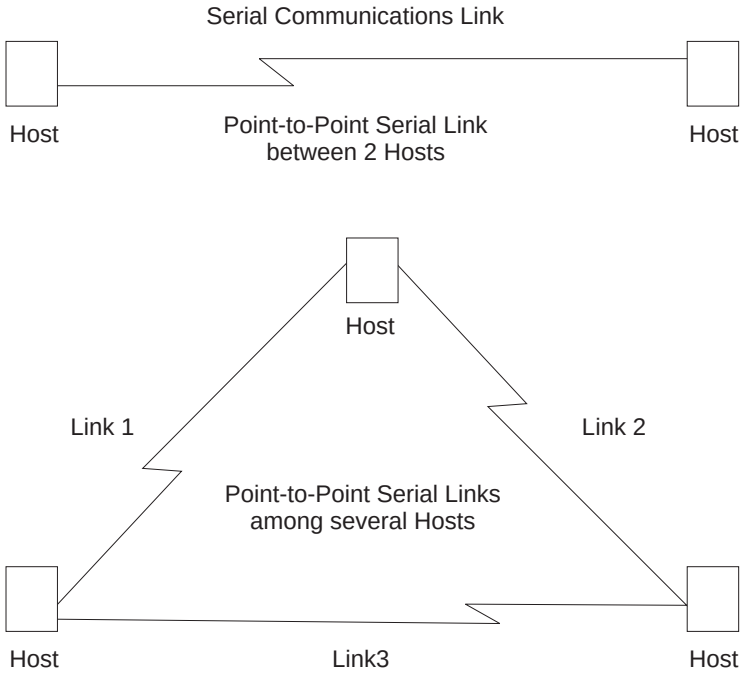



*Figure 44. Examples of Point-to-Point Links*

PPP currently supports AppleTalk Control Protocol (ATCP), DECnet Protocol Control Protocol (DNCP), Banyan VINES Control Protocol (BVCP), bridging protocols (BCP, NBCP, and NBFCP), 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 45 illustrates the PPP frame structure and is followed by a detailed description of each field.

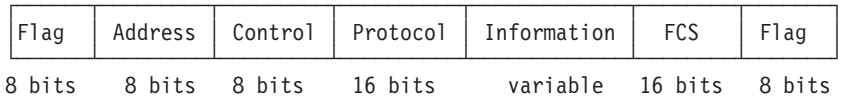



*Figure 45. PPP Frame Structure*

**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.

**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. The 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,

**Information Field**

The information field contains the datagram for the protocol specified in the protocol field. This is zero or more octets.

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).

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.

## 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.
3. After authentication is negotiated, the peers negotiate encryption for the link. After authentication phase is complete, the router negotiates the use of encryption using Encryption Control Protocol (ECP) packets where each end of the link negotiates which encryption algorithm will be used to encrypt the data over this PPP link. If ECP did not reach "Open" state then the link is marked "down" and cannot carry any network traffic. If ECP successfully reaches "Open" state, or if encryption is not required, the PPP link moves to the next phase, NCP negotiation (except ECP, which is technically also an NCP). The link is considered to be "open" or "up" at this time, though it cannot yet carry layer-3 protocol datagrams.
4. 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 happen 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.

5. 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 for other reasons such as: an administrative closing of the link, idle timer expiration, or failure to re-authenticate on a CHAP rechallenge.

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 46 illustrates the structure of the LCP packet and is followed by a detailed description of each field.

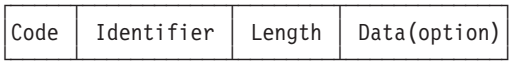



*Figure 46. LCP Frame Structure (in PPP Information Field)*

**Code** The code field is one octet in length and identifies the type of LCP packet. The codes in Table 69 distinguish the packet types. They are described in more detail in later sections.

*Table 69. 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 to 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 sends an appropriate reply, depending on whether it is ready to process packets.

**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 = acknowledgment), 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. On remote dial-in access ports, a third authentication protocol is available. This is SPAP (Shiva Password Authentication Protocol), which is a Shiva proprietary protocol. See “Shiva Password Authentication Protocol (SPAP)” on page 518 for more information.

Whether a box requires the other end to authenticate itself (and if so, with what protocol) is determined during the LCP negotiation phase. 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 on a link, the router initially attempts to use them in the priority order that you specify:

1. CHAP
2. PAP
3. SPAP

**Note:** SPAP is only available on interfaces that have IBM DIALs Dial-In circuits configured.

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.

## Shiva Password Authentication Protocol (SPAP)

**Note:** SPAP is only available on interfaces that have IBM DIALs Dial-In circuits configured.

The Shiva Password Authentication Protocol (SPAP) provides a simple method for the peer to establish its identity using a 2-way handshake similar to PAP. After the Link Establishment phase is complete, an Id/Password is repeatedly sent by the peer to the authenticator until authentication is acknowledged, the connection is terminated, or a retry counter expires.

SPAP is a moderately strong authentication protocol that uses a proprietary encryption algorithm for the password. It offers additional function in concert with authentication:

- The ability to change a password.

  **Note:** SPAP change password support is not available on the 2216
- The ability for the router to send a configurable banner requiring acknowledgment from the client after password authentication.
- The ability to use callback as an additional security feature.

## Configuring PPP Authentication

The following sections describe configuring PPP authentications for two situations:

- Configuring the 2216 to authenticate a remote device.
- Configuring the 2216 to be authenticated by a remote device.

These two situations are independent. You can do one or the other.

## Configuring a PPP Interface to Authenticate a Remote Device

To authenticate a remote device or dial-in client:

1. Enable authentication on the PPP interface
   - At the `Config>` prompt, enter the **network** command to select the PPP interface to configure.
   - At the `PPP Config>` prompt, enable the authentication protocol you want to use.

     You can use any of the following protocols:
     - PAP
     - CHAP
     - SPAP

     **Note:** SPAP is only available on interfaces that have IBM DIALs Dial-In circuits configured.
2. Decide whether to authenticate locally or through an authentication server.
   - To authenticate locally, enter the name and password into the PPP user database.

     At the `Config>` prompt, use the **add ppp_user** command. See “Add” on page 68 for more information.

     A 2216 maintains a single PPP user database. When the remote router or device sends its name and password to the device during the authentication phase, the device checks to see if that name and password are in the PPP user database.
   - To authenticate through an authentication server using TACACS, TACACS+, or RADIUS, you must configure the device to reach the authentication server and the name and password must be in the server’s database. Refer to “Chapter 68. Using Local or Remote Authentication” on page 817.

## Configuring a PPP Interface to be Authenticated by a Remote Device

To configure the device to be authenticated by a remote device or dial-in client, configure the device’s name and password:

1. At the `Config>` prompt, select the interface you are configuring using the **network** command.
2. At the PPP `Config>` prompt, type the **set name** command and provide the name and password that the device will use to identify itself to the remote router or device during the authentication phase.

**Attention:** Do not use the following commands unless you want the device to perform authentication as described in “Chapter 68. Using Local or Remote Authentication” on page 817.

- **enable pap**
- **enable chap**
- **enable spap**

**Note:** SPAP is only available on interfaces that have IBM DIALs Dial-In circuits configured.

# Configuring PPP Callback

Callback is a PPP feature associated with single user dial-in solutions. It attempts to accomplish two objectives. These objectives are:

- Callback can be used as a form of security. When used in this way, callback is generally referred to as required callback. When required callback is negotiated the user will be dialed back at a predetermined number. Only then will the PPP link be allowed to come up.
- Callback can also be implemented as a toll-saver feature. When used in this way, callback is generally referred to as roaming callback. Unlike required callback, roaming callback is requested by the client. The primary function of roaming callback is to bill the company maintaining the DIALs Server the toll charges instead of the user.

Callback is supported only on dial-in dial circuits over ISDN networks.

### Example 1: Required callback enabled

```
Config>add PPP
Enter user name: []? sallydoe
Password:
Enter password again:
Is this a Single-User or a Network? (Single-User, Network): [Single-User]

IP address for user sallydoe [0.0.0.0]?
Enter HostName: []?
Give 'sallydoe' default time allotted ? (Yes, No): [Yes]
Enable Callback for 'sallydoe' ? (Yes, No): [No] yes
Type of Callback (Roaming Callback, Required Callback): [Roaming Callback] Requ
Dialback number for this user []? 555-1234
Will 'sallydoe' be able to dial-out ? (Yes, No): [No]
Enable encryption for this user/port (y/n) [No]:

PPP User Name: sallydoe
Type: Single User
User IP Address: Interface Default
SubNetMask: 255.255.255.255
Hostname: <undefined>
Time-Alotted: Box Default
Call-Back Type: Required Callback
Phone Number: 543-3186
Dial-Out: Not Enabled
Encryption: Not Enabled

Is information correct? (Yes, No, Quit): [No] yes
```

### Example 2: Callback disabled

```
Config>add PPP
Enter user name: []? sallydoe
Password:
Enter password again:
Is this a Single-User or a Network? (Single-User, Network): [Single-User]

IP address for user sallydoe [0.0.0.0]?
Enter HostName: []?
Give 'no callback' default time allotted ? (Yes, No): [Yes]
Enable Callback for 'no callback' ? (Yes, No): [No]
Will 'no callback' be able to dial-out ? (Yes, No): [No]
Enable encryption for this user/port (y/n) [No]:


PPP User Name: no callback
Type: Single User
User IP Address: Interface Default
SubNetMask: 255.255.255.255
Hostname: <undefined>
Time-Alotted: Box Default
Call-Back Type: Not Enabled
Dial-Out: Not Enabled
Encryption: Not Enabled

Is information correct? (Yes, No, Quit): [No] yes
```

### Example 3: Roaming callback enabled

```
Config>add PPP roaming_callback
Password:
Enter password again:
Is this a Single-User or a Network? (Single-User, Network): [Single-User]

IP address for user roaming_callback [0.0.0.0]?
Enter HostName: []?
Give 'roaming_callback' default time allotted ? (Yes, No): [Yes]
Enable Callback for 'roaming_callback' ? (Yes, No): [No] yes
Type of Callback (Roaming Callback, Required Callback): [Roaming Callback]

Will 'roaming_callback' be able to dial-out ? (Yes, No): [No]n
Enable encryption for this user/port (y/n) [No]:


PPP User Name: roaming_callback
Type: Single User
User IP Address: Interface Default
SubNetMask: 255.255.255.255
Hostname: <undefined>
Time-Alotted: Box Default
Call-Back Type: Roaming Callback
Dial-Out: Not Enabled
Encryption: Not Enabled

Is information correct? (Yes, No, Quit): [No]yes
```

## Using AAA with PPP

See “Chapter 68. Using Local or Remote Authentication” on page 817 and “Chapter 69. Configuring Authentication” on page 823 for this information.

## 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 protocols (BCP, NBCP, and NBFCP),
- 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 3 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 the IEEE 802.5 Line Identification Option and the Tinygram Compression Option.

NetBIOS Control Protocol (NBCP) is a proprietary NCP developed by Shiva Corporation and used by the IBM Dial In Access to LAN Client for OS/2, DOS and Windows for single-user dial-in. NBCP is used to transport NetBIOS and LLC/802.2 bridged traffic from these clients, dialed into a 2216 DIALs Server, onto an attached LAN. IBM's implementation of NBCP supports the MAC-Address and NetBIOS Name Projection options.

NetBIOS Frame Control Protocol (NBFCP) is specified in RFC 2097. NBFCP is used by Microsoft Windows 95 and Windows NT Dial-Up Networking clients for single-user dial-in. NBFCP is used to transport NetBIOS bridged traffic from these clients, dialed into a 2216 DIALs Server, onto an attached LAN. IBM's implementation of NBFCP supports the Name-Projection, Peer-Information and IEEE-MAC-Address-Required options.

## 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 following options:

- Van Jacobsen IP Header Compression as described in RFC 1144.
- IP Address

  The router can send its IP address, as well as accept an IP address, from a peer, or supply an IP address to a peer, if requested. If the router is configured to "Send Our Address" on a particular interface, and that interface has a valid, numbered IP address, then IPCP sends the address in its initial Configure-Request as option 3 (IP Address). IPCP also sends its address if the peer sends a Configure NAK with 0.0.0.0 for option 3 (IP Address), if a valid numbered address is configured for that PPP interface. IPCP will not send an unnumbered address to its peer.

  A peer may specify its address (referred to as "Client Specified"), or request an address from the router by sending 0.0.0.0 for Option 3 in its initial Configure Request. The router may obtain this address from the authenticated user profile or from the interface itself. The user profile address takes precedence over the interface address. If you do not want to offer an address from the user profile, simply leave the address for that user in the profile as 0.0.0.0, and the router will offer the remote address configured for that interface. If there is no remote address configured for the interface or user profile, and the peer continues to request an address, IPCP will fail.

The router automatically adds a static route directed to the PPP interface for the address that is successfully negotiated, allowing data to be routed properly to the dial-in client. When the IPCP connection is ended for any reason, this static route is subsequently removed. By default, the net mask for this route is 255.255.255.255 (hostroute), however if a net mask is specified in the authenticated user's profile (see "Configuring PPP Authentication" on page 518) a net mask other than this may be used to allow routing to more than a single host across the PPP link (RIP or other routing protocols could also be used to discover routes if desired).

## 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.

| See "Chapter 70. Overview of Encryption" on page 843 for information about
| configuring encryption for a PPP interface.

# Chapter 42. Configuring and Monitoring Point-to-Point Protocol Interfaces

This chapter describes Point-to-Point Protocol interface configuration and operational commands in the device. Sections in this chapter include:



## 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 9 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 4. The OPCON Process and Commands" on page 29.) 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
```

## Accessing the PPP Interface 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 70 summarizes the PPP configuration commands, and the rest of this section explains these commands. Enter the commands at the `PPP config>` prompt.

*Table 70. Point-to-Point Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Disable | Disables data compression (CCP), DTR line handling, CHAP, PAP, ECP. |
| Enable | Enables data compression (CCP), DTR line handling, CHAP, PAP, ECP. |
| 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 you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Disable

Disables data compression, authentication protocols, multilink PPP, and the Lower DTR feature.

**Syntax:**

**disable** ccp
chap
ecp
lower-dtr
mp
pap

**ccp** Disables the use of data compression on the interface. Refer to “Chapter 66. Using the Data Compression Subsystem” on page 801 for more information.

**chap** Disables the use of the Challenge-Handshake Authentication Protocol. Refer to “Challenge-Handshake Authentication Protocol (CHAP)” on page 518 for more information.

**ecp** This allows the router not to force the use of encryption on this interface. The interface will still accept and execute Encryption Control Protocol (ECP) if the peer is using ECP.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**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) and the interface is disabled, the DTR signal is not dropped.

**mp** Disables the Multilink Protocol (MP) on this interface. See "Chapter 43. Using the Multilink PPP Protocol" on page 561 for more information.

**Example:**

```
disable mp
Disabled as a MP link
```

**pap** Disables the use of the Password Authentication Protocol. Refer to "Password Authentication Protocol (PAP)" on page 518 for more information.

**spap** Disables the use of the Shiva Password Authentication Protocol (SPAP).

**Note:** SPAP is only available on interfaces that have IBM DIALs Dial-In circuits configured.

## Enable

Enables data compression, encryption, authentication protocols, lower-DTR, and the multilink PPP protocol on this PPP interface. If multiple authentication protocols are enabled, the device attempts to use them in the following priority order:

1. CHAP
2. PAP

**Syntax:**

**enable** ccp
chap
ecp
lower-dtr
mp
pap

**ccp** Enables the use of data compression on the interface. See "Chapter 66. Using the Data Compression Subsystem" on page 801 for more information.

**Note:** It is not recommended that you enable data compression for a PPP interface on a HSSI adapter.

**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 518 for more information.

**Example:**

```
enable chap
Rechallenge Interval in seconds (0=NONE)  [0] 10
CHAP enabled
```

**ecp** Enables the use of data encryption on this interface by negotiating Encryption Control Protocol (ECP). Once this is done, all PPP users with encryption enabled and with a valid encryption key must use ECP to connect to this port. PPP users without encryption enabled will still be able to connect to this interface.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**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) and the interface is disabled, the DTR signal is not dropped.

If Lower DTR is set to “enabled”, then the DTR signal will be dropped 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.

**mp** Enables the Multilink Protocol (MP) on this interface. See “Chapter 43. Using the Multilink PPP Protocol” on page 561 for more information.

**Example:**

```
enable mp
Enabled as a MP link
Is this link a dedicated MP link? [no] yes
MP interface for this MP link? [0] 3
```

**pap** Enables the use of the Password Authentication Protocol. Refer to “Password Authentication Protocol (PAP)” on page 518 for more information.

## List

Use the **list** command to display information related to the PPP interface and its protocol parameters and options.

**Syntax:**

**list**

all

bcp

ccp

ecp

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. For additional information, see “Chapter 66. Using the Data Compression Subsystem” on page 801 .

**ecp** Displays the current Encryption Control Protocol state.

**Example:**

```
list ecp
 ECP Options
 -----------
 Data Encryption enabled
 Algorithm list: DESE-CBC
 DESE (Data Encryption Standard Encryption Protocol)
```

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**Data Encryption Enabled/Disabled**
Indicates whether data encryption is enabled or disabled on interface.

**Algorithm List**
Displays the supported encryption algorithms. DES, as described by RFC 1969, is the only encryption algorithm currently supported.

**hdlc** Lists parameters related to the High-Level Data Link Control (HDLC) protocol. 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 53. Using Dial Circuits” on page 653.

**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.

**Notes:**

1. The **list hdlc** command is not supported on PPP dial circuit interfaces.
2. This command displays the Lower DTR state only if Lower DTR is supported for the configured cable type.
3. This command for a PPP interface on a HSSI adapter displays a subset of the HDLC parameters listed above.

**ipcp** Lists the Internet Protocol control protocol options.

**Example:**

```
list ipcp
IPCP Options
------------
IPCP Compression:                 None
Send Our IP Address:               Yes
Remote IP Address to Offer if Requested: 10.0.0.3
```

**IPCP compression**
Indicates whether the PPP handler accepts compressed IP headers. 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 header compression is supported. A value of “NONE” indicates that compressed headers are not being accepted.

**Send Our IP Address**
Indicates where IPCP is configured to send the local IP address for this PPP interface to the remote end of the link in our initial “Configure Request”. Some PPP implementations require this information.

**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
Peer to Local (Rx) ACCM:    A0000
Protocol Field Comp (PFC)      No   Addr/Cntl Field Comp(ACFC)    Yes
```

```
Authentication Options
-------------- -------
Authenticate 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.

**Peer to Local (Rx) ACCM**
Displays the characters that the peer must "escape" when transmitting packets to the router on asynchronous lines.

**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.

**Authenticate 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 223 for an explanation of each of these commands.

**Syntax:**

**llc**

# 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.

**Notes:**

1. 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.
2. The **set hdlc** commands are not supported on PPP dial circuit interfaces.

**Syntax:**

**set** bcp
ccp options
ccp algorithms
hdlc...
ipcp
lcp...
name
ncp...

**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. For additional information, see “Chapter 66. Using the Data Compression Subsystem” on page 801.

**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 nonzero value means that the compression engine maintains the specified number of histories 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 discarded 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, because 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 re-synchronize the sender’s and receiver’s histories.

**Note:** Failure to detect a bad packet can cause all subsequent data to be decompressed incorrectly.

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, out-of-sequence, or corrupted. 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 exclusive–OR checksum. *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.

**4** EXT: An extended mode that is similar to the sequence number mode, in that each packet includes a sequence number, but the compressed frame format is altered more radically. In extended mode, re-synchronization with a peer is performed differently than with the other modes; the signaling between the two nodes is based upon flags passed in the headers of compressed datagrams rather than distinct CCP control packets.

Extended mode is provided for compatibility with certain non- RFC-compliant implementations. It should be used only with clients that do not support mode 3.

**ccp algorithms *list-of-algorithms***

Specifies an exact list of compression protocols to use. The order of preference depends on the order of entry in the list.

When the link negotiates compression with another node, it offers the entire list of protocols to the peer node in preference order. The peer node should select the first protocol it can use from the preference list. Enabling multiple protocols allows the peer to dictate which compression algorithm will be used on the link. If you need to avoid an algorithm, do not specify the algorithm in the list.

Specifying **none** disables the use of any protocol effectively disabling compression. The valid compression algorithms are:

**STAC-LZS**
The STAC-LZS algorithm as described in RFC 1974

**MPPC** The Microsoft Point-to-Point Compression algorithm as described in RFC 2118.

**Example:**

```
set ccp protocols
Enter a prioritized list of enabled compressors
(first is preferred), all on one single line.
Choices (can be abbreviated) are:
Stac-LZS, MPPC
Compressor list [Stac-LZS:]?
```

**hdlc cable** *cable type*

Set the HDLC cable type (that is connected to the interface) to one of the following types:

RS-232 DTE
RS-232 DCE
V35 DCE
V35 DTE
V36 DCE
V36 DTE
X21 DCE
X21 DTE
HSSI DCE
HSSI DTE

Table 71 lists the cable types you can configure on the various adapters.

*Table 71. 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 |
| 1-port HSSI | HSSI DCE and HSSI DTE* |

***Note::** When a HSSI DCE cable is used, the other device must also be configured to use a HSSI DCE cable.

**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 72 on page 536 to determine the clock speeds you can set for the various adapters when internal clocking is used.

**Example:** `set hdlc clocking internal`

**Note:** **Clocking** is set to *external* if the cable type is *HSSI DTE* and is set to *internal* if the cable type is *HSSI DTE* and is not configurable.

**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.

**Note:** Encoding is set to NRZ for a PPP interface on a HSSI adapter and is not configurable.

**Example:** `set hdlc encoding nrz`

**hdlc idle** *flag* **or** *mark*

Sets the data link idle state to either Flag or Mark.

The flag option provides continuous flags (7E hex) between frames.

The mark option puts the line in a marking state (OFF, 1) between frames.

**Note:** Idle is set to *flag* for a PPP interface on a HSSI adapter and is not configurable.

**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 72.

*Table 72. 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 |
| 1-port HSSI | 22 368 000 bps or 44 736 000 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 73 to determine the clock speeds you can set for the various adapters.

*Table 73. 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 |
| 1-port HSSI | 1 544 000 bps to 52 000 000 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.

**Note:** If you configure a nonzero transmit delay for a PPP interface on the 8-port EIA- 232E adapter, 6-port V.35/V.36 adapter, or 8-port X.21 adapter, you must configure the line speed using the **set hdlc speed** command.

**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]:
Note: unnumbered interface addresses will not be sent.
Interface remote IP address to offer if requested (0 for none) [0.0.0.0]? 10.0.0.3
```

**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.

If set to “yes”, IPCP will send the IP address of the PPP interface, if the interface is configured with a numbered IP address, (That is, the address does not begin with 0). If this option is set to “no” and the peer sends us a Configure NAK with 0.0.0.0 for the IP Address option, the 2216 will respond with the address of the PPP interface if it is configured with a numbered address.

**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]?
```

**Maximum receive unit**

Sets the maximum 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 bit mask 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.

**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** *routerid key*

Sets the name that the router uses when responding to authentication requests from another router. Also sets the device's encryption key.

**Notes:**

1. 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.
2. Other implementations may not handle name and passwords with the same maximum length as supported in this product. The only indication would be a message from the authenticator stating that there is a bad name or password. If you receive this type of message, try shortening the routerid and key.

You will be prompted to enter the encryption key as 16 hexadecimal characters.

**Example:**

```
set name routerid key
Config>
Config>net x
PPP x Config>
PPP x Config>set name
Enter Local Name:  []?newyork
Password:
Enter password again:
Enable encryption for this user/port (y/n) [No]:y
Encryption key should be 16 characters long.
Encryption Key (16 characters ) in Hex(0-9, a-f, A-F):
Encryption Key again (16 characters) in Hex(0-9, a-f, A-F):
PPP Local Name = newyork
PPP x Config>
```

**ncp parameters**

Sets the basic operational parameters for most NCPs.

**Note:** Although you access this command through a particular interface, this command will reset the parameters for all PPP interfaces.

**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.

## Accessing the Interface Monitoring Process

To access the PPP interface monitoring 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 Monitoring Commands

This section summarizes and then explains the Point-to-Point monitoring commands. Enter the commands at the PPP> prompt. Table 74 shows the commands.

**Note:** The options available for these commands depend on 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 74. Point-to-Point Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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 monitoring prompt. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Clear

Use the **clear** command to clear all statistics from point-to-point interfaces.

**Syntax:**

**clear**

## List

Use the **list** command to display information and counters related to the point-to-point interface and PPP parameters and options.

**Syntax:**

**list**

all
control
errors
interface
lcp - PPP link CP
pap - PAP Authentication CP
chap - CHAP Authentication CP
ecp - Encryption Control Protocol
edp- Encrypted packet statistics
ccp - PPP Compression CP
cdp - PPP compression
compression - PPP compression
bcp - Bridging (ASRT) CP
brg - Bridging (ASRT)
stp - Spanning Tree Protocol
ipcp - Internet Protocol CP
ip - Internet Protocol
ipxcp - Novell IPX CP
ipx - Novell IPX
atcp - AppleTalk (Phase 2) CP
ap2 - AppleTalk (Phase 2)
dncp - DECnet IV CP
dn - DECnet IV
osicp - ISO's OSI CP
osi - ISO's OSI
bvcp - Banyan VINES CP
vines - Banyan VINES
isrcp - APPN ISR CP

isr - APPN ISR

hprcp - APPN HPR CP

hpr - 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 displays from all of the individual **list** *item* 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.

**control**

Lists negotiated options or other state information for a control protocol.

ccp

ecp

lcp

bcp

nbcp

nbfcp

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**

The current state of the point-to-point link. If "Open" then compression was successfully negotiated on this link. If not open, compression is not running on the 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:**

```
PPP x>list control ecp

ECP State:                Open
Previous State:           Ack Sent
Time Since Change:        16 minutes and 40 seconds

Local (transmit) encrypter: DES
Remote (receive) encrypter: DES
```

**ECP State:**
The current state of the point-to-point link. If “Open” then encryption was successfully negotiated on this link. If not “Open”, encryption is not running on the link.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88 .

**Previous State:**
The state of the point-to-point link before the state displayed in the current state field.

**Time Since Change:**
The elapsed time between the above two state changes.

**Local (transmit) encrypter:**
This encryption algorithm is used for encrypting the data being sent on this PPP interface.

**Remote (receive) encrypter:**
The encryption algorithm is used for decrypting the received data on this interface.

**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.

**Link phase**
Displays the current activity on the link. This can have one of the following values:

**Dead** There is no activity on the link; the interface is down.

**LCP** The link is in LCP negotiation. This state occurs when first bringing up an interface. The interface may be in self-test at this time.

**Authenticate**
The link is performing initial authentication.

**ECP** The link is negotiating an encryption algorithm.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**Ready** Link is operating normally. NCPs can negotiate and data traffic associated with can flow after successful NCP negotiation.

**Terminate**
The link is being shut down.

**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 can be sent. The retry timer does not run in this state.

CLOSED - Indicates that the link is down and no attempt is being made to open it. In this state, all connection requests from peers are rejected.

LISTEN - Indicates that the link is down and no attempt is 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 is an "Identification" packet. The contents of this packet are undefined but are normally expected to be a human-readable string provided by the peer to give 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 are 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 the **set lcp options** command on page 537 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 nbfcp
NBFCP State:               Closed
Previous State:            Closed
Time Since Change:         4 hours, 5 minutes and 58 seconds

NetBIOS Frame Control Protocol Info:
Local  MAC Address = 0x000000000000
Remote MAC Address = 0x444553540000
Remote NetBIOS Names: (0)

Remote Peer Class:         0
Remote Peer Version Major: 0
Remote Peer Version Minor: 0
```

The NBFCP State fields are the same as those described under the **list control lcp** command.

**Local MAC Address**
The Local MAC Address is the MAC Address that is used by the Win 95/NT Dial-Up Networking client. It is a pseudo-random number, or a Locally Administered Address (LAA), if you configured an LAA in the client.

**Remote MAC Address**
The Remote MAC Address is the MAC Address that the 2216 DIALs Server has assigned to this client for use on the LAN.

**Remote NetBIOS Name**
The list of NetBIOS names of LAN resources to which the client has requested access.

**Remote Peer**
The Remote Peer Class, Version Major, and Version Minor is the information passed back to the 2216 by the NBFCP Peer Information option.

**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             10.0.0.152
Compression Slots          None                None

DHCP State:                BOUND
Lease Server:              10.0.0.111
Leased IP Address:         10.0.0.152
Lease Time:                4 minutes and 0 seconds
Renewal Time:              2 minutes and 0 seconds
Rebind Time:               3 minutes and 30 seconds
Lease Time Elapsed:        1 second
Lease Time Remaining:      3 minutes and 59 seconds

DHCP Client ID:            0100120B0000
```

The IPCP state fields are the same as those described under the **list control lcp** command.

**IP Address:**
Indicates if this interface's IP address (Local) and the negotiated address of the remote (Remote), if any.

**Compression Slots**
Indicates the number of IP headers saved for referential purposes when determining the type of compression that is enabled.

**DHCP State**
This is the Proxy DHCP as described in RFC 1541.

**Lease Server**
The server from which the lease was acquired.

**Leased IP address**
The address leased to the client. This address should be equivalent to the "Remote IP Address" listed above.

**Lease Time**
Length of lease from the DHCP server for this address. When "Lease Time Elapsed" equals this time, the lease will be expire and the IPCP connection closed.

**Renewal Time**
Time after which Proxy DHCP attempts to extend this lease from the server. When "Lease Elapsed Time" equals this time, Proxy DHCP attempts to renew the lease, resetting the "Lease Time," "Lease Elapsed Time," and "Lease Time Remaining," if successful.

**Rebind Time**
Time before Proxy DHCP attempts to obtain a new lease from any configured DHCP server. When "Lease Elapsed Time" equals this time, Proxy DHCP attempts to obtain a new lease, resetting the "Lease Time," "Lease Elapsed Time," and "Lease Time Remaining," if successful.

**Leased Time Elapsed**
Time elapsed for this lease. This is not necessarily the time for this particular dial-in session, as the lease may have been renewed. When the lease is renewed, this timer is set back to 0.

**Leased Time Remaining**
Time remaining for this lease. This parameter is equal to “Lease Time” minus “Lease Time Elapsed.”

**DHCP client ID**
A unique ID for this client (dial-in user). All DHCP messages are identified to and from the DHCP server by this client ID.

**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
APPN ISRCP 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 interface.

**Reset Reqs**
The number of CCP dictionary “Reset Requests” that were transmitted or received.

**Reset Acks**
The number of CCP dictionary “Reset Acknowledgments” that were transmitted or received.

Reset Request and Reset Acknowledgment 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 associated with compressed data packets sent or received on this interface.

**Example:**

```
list cdp
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
```

**Packets**

These counters indicate the number of compressed datagrams sent and received. On the output side, the count includes only those packets that were actually sent as PPP compressed datagrams; it does not include packets that were found to be incompressible and sent in their original uncompressed form.

These counters count the packets sent or received that had the PPP protocol type of X'00FD' (CDP). When STAC extended mode or MPPC has been negotiated, incompressible packets may be encapsulated in CDP datagrams. This encapsulation would include the incompressible packets in these counts.

**Octets**

These counters indicate the number of bytes effectively transmitted or received in compressed form. These counts reflect the lengths of the original datagrams before compression or after decompression.

**Compressed octets**

These counters indicate the number of bytes for all of the compressed datagrams sent and received. These counts are the lengths of the actual CDP packets after compression or before decompression.

**Incompressible packets**

These counters indicate the number of packets that were incompressible and therefore sent in original uncompressed form.

**Discarded packets**

These counters indicate how many packets were discarded because they could not be successfully decompressed. Typically these packets will be packets that the peer was transmitting just after the router has sent a Reset-Request, but before the peer has received and processed the Reset-Request. Packets are also dropped if the router detects that data in the packets is incorrect. An example of incorrect data is a packet that contains a bad sequence number.

If the number of discarded packets increases too rapidly, then packets are being lost or corrupted on the line, probably due to noise on the line, and the link performance may be degraded.

**Protocol rejects**

This counter indicates the number of Protocol-Rejects of CDP packets that have been received from a peer. This count should be zero, because the link will not send CDP packets if the use of compression has not already been negotiated.

**Compression ratios**

The ratios give an approximate indication of the effectiveness of the compressor and decompressor. These ratios are based on the number of plain-text bytes divided by the number of corresponding

compressed bytes, so values greater than 1 are preferable for both input and output. The higher the number, the more effective the compression.

The output ratio is computed as the ratio of the number of original plain-text bytes divided by the number of bytes sent as a result of attempting compression - whether the packet actually was compressed or sent as a CDP packet. If a data stream does not compress well and most of the packets are sent in their original form or in enlarged CDP packets, the compression output ratio will drop. If the ratio drops below 1.0, the compressor is actually reducing the effective bandwidth of the line rather than increasing it, and should be disabled on that interface if the state persists for a long time.

The input ratio is computed based on the number of bytes received in CDP frames divided into the number of decompressed bytes. Unlike the output ratio, this count does not include any packets that were incompressible and sent in plain-text form. This is because the router cannot determine if a received non-CDP packet was an incompressible packet that the peer sent in plain-text form, or just a packet that the peer did not attempt to compress.

Because of the method of calculation, the output ratio on one end of the link does not necessarily match the input ratio at the other end.

**compression**

This command displays the same information as list cdp.

**ecp** Lists statistics for encryption control protocol packets sent or received on the interface.

**Example:**

```
PPP x>list ecp

ECP Statistic              In                      Out
-------------              --                      ---
Packets:                   2                       2
Octets:                    26                      26
Reset Reqs:                0                       0
Reset Acks:                0                       0
Prot Rejects:              0                       -
Local (transmit) encrypter: DES
Remote (receive) encrypter: DES
```

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**Packets**

Indicates the total number of ECP packets transmitted (out) and received (in) over the current point-to-point interface.

**Octets**

Indicates the total number of bytes transmitted and received in the ECP packets.

**Reset Reqs**

Indicates the number of Reset requests transmitted and received on this interface. A Reset Request will be sent whenever ECP discard an EDP packet.

**Note:** Because DES, the only supported encryption algorithm, does not send reset requests this number will be zero.

**Reset Acks**
Indicates the reset acknowledgments transmitted and received on this interface. A Reset Ack packet will be sent for every Reset Request packet received.

**Note:** Because DES, the only supported encryption algorithm, does not send any Reset Requests this number will be zero.

**Prot Rejects**
Indicates the total number of protocol reject packets transmitted and received over the current point-to-point interface.

**Local (transmit) encrypter**
This encryption algorithm will be used to encrypt the data being sent on this point-to-point interface.

**Remote (receive) encrypter**
This encryption algorithm will be used to decrypt the received data on this point-to-point interface.

**edp** Lists statistics associated with the encrypted packets being sent or received on the interface.

**Example:**

```
PPP x>list edp

Encryption Statistic       In                    Out
--------------------       --                    ---
Packets:                   20                    30
Octets:                    29164                 44790
Encrypted Octets:          29280                 44880
Discarded Packets:         0                     0
Prot Rejects:              0                     -
```

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

**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.

**Encrypted Octets**
Indicates the number of encrypted octets transmitted or received on this interface.

**Discarded Packets**
Indicates the number of packets that were discarded because they could not be successfully decrypted.

**Prot Rejects**
Indicates the total number of protocol reject packets transmitted and received over the current point-to-point interface.

**bcp** Lists statistics for the Bridging control protocol. These fields are the same as those described under the **list ip** command. (See 557.)

**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 557.)

**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 557.)

**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 557.)

**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 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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**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 557.)

**Example:**

```
list isrcp
APPN ISRCP 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 557.)

**Example:**

```
list isr
APPN ISR 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 557.)

**Example:**

```
list hprcp
APPN HPRCP 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 557.)

**Example:**

```
list hpr
APPN HPR 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 227 for an explanation of each of these commands.

**Note:** This command is available only when APPN is included in the software load.

**Syntax:**

**llc**

# Point-to-Point Protocol Interfaces and the GWCON Interface Command

The PPP interface traffic is carried by an underlying data-link level device driver. Additional statistics that can be useful when monitoring PPP links may be obtained from the device driver statistics which are displayed using the **interface** command from the GWCON environment. (For more information on the **interface** command, refer to “Chapter 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands” on page 99.)

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
```

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 43. Using the Multilink PPP Protocol

The Multilink PPP Protocol (MP) allows you to increase the bandwidth of ISDN B-channels by defining a ***virtual link*** made up of multiple links. The bandwidth of the resulting MP bundle is almost equal to the sum of the bandwidths of the individual links. The advantage is that large data packets transmitted across a single link can now be fragmented, transmitted across multiple links, and rebuilt at the receiving end station. MP helps eliminate bottlenecks in the ISDN portion of your network. MP uses both the Bandwidth Allocation Protocol and the Bandwidth Allocation Control Protocol to, add links to and drop links from, a virtual link.

There are two types of MP links: those that are dedicated and those that are simply enabled. A dedicated MP link is an MP-enabled dial circuit configured as a link to a particular MP interface. If the dial circuit attempts to join another MP bundle, or if MP is not negotiated at all, the software ends the call. An MP-enabled dial circuit that is not dedicated can become a link in any MP bundle. If MP is not negotiated, the dial circuit operates as an independent interface using the dial circuit's configured protocols.

**Important:** You cannot use a dial circuit that has a Channelized ISDN T1/E1 interface as its base net as part of an MP bundle.

You can configure an Multilink PPP interface that consists of multiple PPP dial circuits as part of the MP bundle. Each of the PPP dial circuit interfaces must use an ISDN base net.

There are also two types of MP interfaces: those that have a dedicated link and those that do not. An MP interface needs a dedicated link in any one of the following situations:

- The link is only for the MP interface
- The MP interface is configured for outbound calls. The dedicated link must then be configured with the destination phone number and caller identification.
- The MP interface is configured to receive a particular inbound call. In this case, the dedicated link is configured with the inbound destination phone number and caller identification.
- The MP interface needs to perform outbound authentication. In this case, all links use the same authentication name.

MP interfaces that do not have a dedicated link must be inbound-only interfaces. These interfaces are similar to the any inbound dial circuit.

The Bandwidth Allocation Protocol (BAP) and its control protocol (BACP) allow an MP interface to increase and decrease its bandwidth by adding and dropping ISDN B-channels. When the bandwidth utilization algorithm determines that a link should be added to the bundle, if there is an available PPP dial-circuit, an available B-channel, and the peer agrees, an additional call is placed.

BAP first searches for any idle dedicated PPP dial circuits for the MP interface, and then for any MP-enabled PPP dial circuit. It will not, however, use a dedicated PPP dial circuit of another MP circuit. The configured maximum number of links on the MP interface will never be exceeded.

## Configuring a Multilink PPP Interface

This section shows how to configure a Multilink PPP interface by using an example that configures Multilink PPP with two ISDN dial circuits.

1. Add the two dial circuits and the multilink PPP interface.

```
*t 6

Config>add dev dial-circuit
Adding device as interface 7
Defaulting Data-link protocol to PPP
Use "net 7" command to configure circuit parameters
Config>add dev dial-circuit
Adding device as interface 8
Defaulting Data-link protocol to PPP
Use "net 8" command to configure circuit parameters
Config>add dev multilink-ppp
Adding device as interface 9
Defaulting Data-link protocol to PPP
Use "net 9" command to configure circuit parameters
Config>
```

2. Configure each PPP dial circuit. (See “Chapter 53. Using Dial Circuits” on page 653 .) In this example, the destination, call direction, and LIDs are set for one of the dial circuits.

```
Config>net 7
Circuit configuration
Circuit config:   7>set dest out
Circuit config:   7>set calls outbound
Circuit config:   7>set net 6
Circuit config:   7>
```

3. Enable MP on each dial circuit to be used for MP as follows:

```
Circuit config:   7>encapsulator
Point-to-Point user configuration
PPP 7 Config>enable mp

Enabled as a Multilink PPP Link,
Use as a dedicated Multilink PPP link? [No]: yes
Multilink PPP net for this Multilink PPP link [1]? 9
NOTE: PPP configuration will be obtained from the Multilink PPP
      net.  It is NOT necessary to configure PPP for this net!
```

   **Note:** You cannot configure PPP parameters for dedicated links from this prompt. Dedicated links use the existing MP interface’s PPP configuration.

   By answering “Yes” to the question “Use as a dedicated Multilink PPP link?” the link becomes dedicated to the specified Multilink PPP interface (9 in this example). In this case, the link ***must*** be used for an MP bundle and ***must*** join the specified MP interface. The link cannot be used as a regular PPP dial circuit.

   Answering “No” to “Use as a dedicated Multilink PPP link?” will allow this PPP dial-circuit to join any MP interface. At least one PPP dial-circuit ***must*** be a dedicated link to an outbound MP interface.

   A dedicated PPP dial circuit obtains all PPP parameters (LCP options, authentication, and others) from its MP interface. MP enabled PPP dial circuits joining the same MP bundle ***must*** negotiate the same LCP parameters and authentication name.

4. Configure the MP interface. The “Dialout MP link net” should be a dedicated PPP dial circuit.

```
Config>net 9
Circuit configuration
MP config:   9>set calls out
Dialout MP link net for this MP Net [0]? 7
MP config:   9>
```

Protocols, BAP, BRS, WAN restoral, WAN reroute, and dial-on-demand are all run on the MP interface and not the PPP dial circuits.

# Chapter 44. Configuring and Monitoring Multilink PPP Protocol (MP)

This chapter describes how to configure specific Multilink PPP interfaces in a device. The chapter includes:



## Accessing the MP Configuration Prompt

To access the `MP config>` prompt:

1. Enter **talk 6** at the `*` prompt.
2. Enter **net *n***, where n is the number of the dial circuit that you enabled to use MP.

   **Note:** You are now configuring the Multilink PPP interface and not the PPP dial circuit that is part of the MP bundle.

## MP Configuration Commands for Multilink PPP Interfaces

Table 75 lists the commands available at the `MP config>` prompt.

*Table 75. MP Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Disable | Disables the negotiation of BAP/BACP and bandwidth on demand. |
| Enable | Enables the negotiation of BAP/BACP and bandwidth on demand. |
| Encapsulator | Places you in the `PPP config>` prompt so you can change the data-link protocol configuration. |
| List | Displays the MP interface configuration parameters. |
| Set | Configures MP interface for inbound or outbound traffic. Also allows you to set the idle timeout and other MP and BAP parameters. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Disable

Use the **disable** command to disable the negotiation of BAP. Disabling BAP prevents the link from allocating additional bandwidth when necessary.

**Syntax:**

**disable** bap

## Enable

Use the **enable** command to enable the negotiation of BAP. Enabling BAP allows the link to allocate additional bandwidth when necessary.

**Syntax:**

**enable** bap

## Encapsulator

Use the **encapsulator** command to access the PPP link-layer configuration for the Multilink PPP interface.

**Syntax:**

**encapsulator**

**Example:**

```
encapsulator
Point-to-Point user configuration
PPP config>
```

## List

Use the **list** command to display the current MP configuration.

**Syntax:**

**list**

**Example:**

```
list
Idle timer = 0 (fixed circuit)
Outbound calls = allowed
Dialout MP Link net = 7
Max fragment size = 750
Min fragment size = 375
Maximum number of active links = 2
Links associated with this MP bundle:
net number 7
net number 8
BAP enabled
Add bandwidth percentage = 90
Drop bandwidth percentage = 70
Bandwidth test interval (sec) = 15
```

**Idle timer**
The setting of the idle timer for this circuit in seconds.

A setting of 0 indicates a fixed circuit. A nonzero setting configures a dial-on-demand MP circuit that will be brought down when the circuit is idle for the specified number of seconds The circuit is reactivated when network traffic resumes.

**Outbound calls**
Specifies whether the interface is configured to initiate outbound calls. If the interface cannot initiate outbound calls, this line is not displayed.

**Inbound calls**
Specifies whether the interface is configured to initiate inbound calls. If the interface cannot accept inbound calls, this line is not displayed.

**Dialout MP link net**
The ISDN dial circuit configured to place the first call for an outbound MP circuit.

**Max fragment size**
Specifies the largest number of bytes of data a packet can contain before the packet is fragmented to be sent over MP links.

**Min fragment size**
This is the minimum size of the fragments (in bytes) the software creates when a packet exceeds ***Max fragment size***.

**Maximum number of active links**
Specifies the configured maximum number of links in the MP virtual link (also known as ***bundle***).

**Links associated with this MP bundle**
Displays the links dedicated to this MP interface.

**BAP enabled**
Specifies whether BAP is enabled on this interface.

**Add bandwidth percentage**
The amount of bandwidth utilization at which the software will try to add a new link if BAP is enabled.

**Drop bandwidth percentage**
The amount of bandwidth utilization at which the software will remove a link from the MP bundle if BAP is enabled.

**Bandwidth test interval**
The time, in seconds, after which the software will check the bandwidth utilization to determine whether to add or drop a link from the bundle.

## Set

Use the **set** command to configure:

- The MP interface for inbound or outbound calls
- The idle timeout
- The MP parameters
- The BAP parameters

**Syntax:**

**set** bap parameters
calls
idle
mp parameters

**bap parameters**
Prompts you to specify the BAP add and drop bandwidth percentages and the BAP test interval.

**Example:**

```
set bap parameters
Add bandwidth % [90]? 80
Drop bandwidth % [70]? 50
Bandwidth test interval (sec) [15]? 25
```

**Add bandwidth %**
The amount of bandwidth utilization at which the software will try to add a new link.

**Valid Values:** 1 to 99

**Default Value:** 90

**Drop bandwidth %**
The amount of bandwidth utilization at which the software will remove a link from the MP bundle.

**Valid values:** 1 to 99

**Default value:** 70

**Bandwidth test interval (sec)**
The time, in seconds, after which the software will check the bandwidth utilization to determine whether to add or drop a link from the bundle.

**Valid Values:** 10 to 200 seconds

**Default Value:** 15

**calls** Specifies whether this MP interface will initiate outbound calls, only accept outbound calls, or participate in both types of calls.

**Valid values:** inbound, outbound, or both

**Default value:** inbound

**Note:** If you specify outbound or both, the software will request the net number of the dedicated MP link that will place the first call.

**Example:**

```
set calls outbound
Dialout MP link net for this MP net []? 4
```

**idle** Specifies the time period in seconds that an interface can have no protocol traffic at which the MP interface will end calls on all the links.

**Valid Values:** 0 to 65535

**Default Value:** 0

**mp parameters**
Prompts you to enter the maximum and minimum fragment sizes and the maximum number of active links.

**Example:**

```
set mp parameters
Max frag size [750]? 675
Min frag size [375]? 300
Max number of active links [2]? 4
```

**Max frag size**
Specifies the largest of number of bytes of data a packet can contain before the packet is fragmented to be sent over MP links.

**Valid Values:** 100 to 3 000

**Default Value:** 750

**Min frag size**
This is the minimum size of the fragments (in bytes) the software creates when a packet exceeds ***Max fragment size***.

**Valid Values:** 100 to 3 000

**Default Value:** 375

**Max number of active links**
Specifies the configured maximum number of links in the MP virtual link (also known as ***bundle***).

**Valid Values:** 1 to 64

**Default Value:** 2

## Monitoring MP Interface Status

To determine the status of all the MP interfaces in your device, use the **configuration** command in **talk 5** (see “Configuration” on page 102).

## Accessing the MP Monitoring Commands

To access the MP monitoring commands:

1. Enter **talk 5** at the * prompt.
2. Enter **net *n***, where ***n*** is the number of the MP interface.

## Multilink PPP Protocol Monitoring Commands

Table 76 shows the monitoring commands available for an MP interface.

*Table 76. MP Monitoring Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Displays BAP, BACP, and MP statistics, errors, and other information. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to display information about the MP interface including bandwidth allocation statistics.

**Syntax:**

**list**

bacp

bap

control bacp

control bap

control mp

mp

**Note:** The examples that follow assume that the MP interface on this device is net number 6.

**bacp** The **list bacp** command lists the statistics for bandwidth allocation control packets which have been sent or received on this MP circuit.

**Example:**

```
PPP 6> list bacp

BACP Statistic            In          Out
--------------            --          ---
Packets:                  6           8
Octets:                   60          80
Rejects:                  0           -
```

**bap** The **list bap** command lists the statistics for bandwidth allocation protocol packets which have been sent or received on this MP circuit.

**Example:**

```
PPP 6> list bap

BAP Statistic               In          Out
-------------               --          ---
Packets:                    3           3
Octets:                     22          37
Call Requests:              1           0
Call Response(ACK):         0           1
Call Resp(NK & FLLNK):      0           0
Call Response(Rej):         0           0
Callback Requests:          0           0
Callback Response(ACK):     0           0
Cllbck Resp(NK & FLLNK):    0           0
Callback Response(Rej):     0           0
Drop Requests:              0           1
Drop Response(ACK):         1           0
Drop Resp(NK & FLLNK):      0           0
Drop Response(Rej):         0           0
Call Status(Success):       1           0
Call Status(Fail):          0           0
```

There are four different responses to a peer's request: ACK, NAK, FULL-NAK, and REJECT.

**ACK** Indicates the peer's request has been granted.

**NAK (NK)**
Indicates that the peer's request is supported but not desired at this time. Try again later.

**FULL-NAK (FLLNK)**
Indicates that the peer s request is supported but because of a resource condition, cannot be granted at this time. The request should not be sent again until the total bandwidth across the MP bundle changes.

**REJECT (REJ)**
Indicates that the request is not supported.

**control bacp**
The **list control bacp** command lists the current state of the BACP state-machine within PPP. The state information is identical to that produced for all of the PPP control protocols. Information about favored peer is also listed. Favored peer is used to alleviate BAP packet collisions (when both sides simultaneously initiate requests). During BACP negotiations, each side sends a magic-number and the one with the smallest magic number is the favored peer and should take precedence in the event of a collision. Typically, the call initiator will choose a ***magic number*** of X'1' and the call receiver will choose a magic number of X'FFFFFFFF' establishing the call initiator as the favored peer.

```
PPP 6> list control bacp

BACP State:                Open

BACP Option                Local                    Remote
----------                 -----                    ------
Magic Number:              FFFFFFFF                 1
Favorite Peer:             NO                       YES
```

### control bap

The **list control bap** command lists the state of the bandwidth allocation protocol and bandwidth on demand. This information includes BAP state, configured bandwidth on demand parameters for adding and subtracting bandwidth, current bandwidth, and information from the last bandwidth poll.

**Example:**

```
PPP 6> list control bap

BAP State:                                      Ready
Bandwidth test interval (sec):                  15
Add bandwidth percentage:                       90
Drop percentage (links-1):                      70
Max # active links in MP bundle:                3
Time since last Bandwidth check (sec):          5
Currently:
    # active links in MP bundle:                1
    Total MP bandwidth (Bytes/sec):             8000
Last Bandwidth Check:
    # active links in MP bundle:                2
    Avg Inbound bandwidth util (%):             12
    Avg Outbound bandwidth util (%):            12
    Drop check: Avg In (%) for links-1:         24
    Drop check: Avg Out (%) for links-1:        24
```

**Note:** Drop percentage considers current utilization for links - 1

Valid BAP states are:

**Closed**
BACP is not opened – BAP either is not enabled or not supported by the peer.

**Ready** BACP is opened and there is no outstanding request being processed.

**Call Req Sent**
There is an outstanding call-request that was sent from the local machine.

**Callback Req Sent**
There is an outstanding callback-request that was sent locally.

**Call Placed**
As a result of a BAP request to add bandwidth, a call has been placed.

**Retry Status Sent**
The outgoing call failed to join the MP bundle, a retry status was sent.

**No Retry Status Sent**
The outgoing call either succeeded or exhausted all retries, a no retry status was sent.

**Drop Req Sent**
There is an outstanding drop request that was sent locally.

Configured bandwidth-on-demand parameters include add percentage, drop percentage, maximum number of active links in the MP bundle, and the bandwidth polling interval.

A BAP request to add a link to the bundle will be initiated if both the following conditions are met:

- The current number of active links is less than the configured maximum number of links.
- The bandwidth utilization across all links in the MP bundle is greater than the add percentage of the total available bandwidth for the MP bundle.

A BAP request to drop a link from the MP will be initiated if both the following conditions are met:

- The number of active links is greater than one.
- The bandwidth utilization across all links in the MP bundle is less than the drop percentage of the total available bandwidth for the MP bundle for the number of links minus one.

Bandwidth can be polled only when BAP is in the ready state. The information listed from the previous poll will give you an idea of the bandwidth utilization across the MP bundle.

These two sets of information are displayed when a drop can be initiated:

- Bandwidth utilization across the entire bundle
- Bandwidth utilization across number of links minus one

To prevent thrashing, the second set of information is used when determining whether to drop a link.

**control mp**

The **list control mp** command lists the current state of this MP circuit including the number of active links and bandwidth, the configured maximum number of links, and statistics for number of dropped packets. Dropped MP packets are classified into four categories:

**M** The packet is dropped because a sequence number has not been received and it is less than the minimum sequence number across all links' last received sequence number.

**Timeout**
The packet is dropped because a sequence number has not been received during a timeout period.

**Q depth**
The packet is dropped because the maximum queue depth was exceeded.

**Seq order**
The packet is dropped because the sequence number received was not expected. This occurs when MP receives delayed packet that it has already declared lost.

If a packet is dropped at the network layer, it can be either an M, Timeout, or Q depth packet. These counters are incremented appropriately when a packet is dropped.

```
PPP 6> list control mp

Current # active links in MP bundle:        2
Max # active links in MP bundle:            3
Total MP bandwidth (Bytes/sec):             16000
Dropped Frags (lost - M):                   0
Dropped Frags (timeout):                    0
Dropped Frags (Q depth):                    0
Dropped Frags (seq order):                  0
```

**mp** The **list mp** command lists the statistics for packets which have been sent or received on this MP circuit. The number of bytes displayed is for pre-decompressed packets if compression was negotiated for the multilink PPP bundle.

```
PPP 6> list mp

MP Statistic               In          Out
------------               --          ---
Bytes (Compressed):        61230       60259
```

# Chapter 45. Using SDLC Relay

This chapter describes how to use the Synchronous Data Link Control (SDLC) Relay interface. The chapter includes the following sections:

- "Basic Configuration Procedure"

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 3 Release 1*.

## 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.

# Chapter 46. Configuring SDLC Relay

This chapter describes the Synchronous Data Link Control (SDLC) Relay configuration and operational commands. The chapter includes the following sections:



## 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>
```

## SDLC Relay Configuration Commands

This section summarizes 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 77 shows the commands for both the **network sdlc** and **protocol sdlc**.

*Table 77. SDLC Relay Configuration Commands Summary*

| | Network | Protocol | |
|---|---|---|---|
| Command | SRLY | 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. |

*Table 77. SDLC Relay Configuration Commands Summary (continued)*

| Command | Network SRLY | Protocol SDLC | Function |
|---|---|---|---|
| Exit | yes | yes | Exits the SDLC Relay configuration environment and returns to the CONFIG environment. |

# Add

Use the **add** command to add group numbers, local ports, and remote ports.

**Syntax: add**

group

local-port

remote-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: delete**

group . . .

local-port . . .

remote–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. The maximum frame size must be large enough to accommodate the largest frame and the 15 byte SRLY header.

**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 78 lists the cable types you can configure on the various adapters.

*Table 78. 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 80 on page 583 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 IBM 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 79.

**Note:** The frame size must be large enough to accommodate the largest frame received plus a 15–byte SRLY header.

*Table 79. Valid Values for Frame Size in Set Frame-Size Command*

| Minimum | Maximum | Default |
|---|---|---|
| 128 | 8187 | 2048 |

**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.

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.

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 80 to determine the link speeds you can set for the various adapters.

*Table 80. 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 81 for the line speeds supported when external clocking is used.

*Table 81. 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 |

**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.

**Note:** If you configure a non-zero transmit delay for a SDLC Relay interface on the 8-port EIA- 232E adapter, 6-port V.35/V.36 adapter, or 8-port X.21 adapter, you must configure the line speed using the **set speed** command.

## Accessing the SDLC Relay Monitoring Environment

To monitor information related to the SDLC Relay interface, access the interface monitoring process by doing the following:

1. Enter the **status** command to find the PID for GWCON. (See page 9 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 102 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 monitoring commands.

## SDLC Relay Monitoring Commands

This section summarizes and then explains the SDLC Relay monitoring commands. The SDLC Relay monitoring commands allow you to view parameters for interfaces transmitting SDLC Relay frames. The `SDLC Relay>` prompt is displayed for all SDLC Relay monitoring commands. Table 82 shows the commands.

*Table 82. SDLC Relay Monitoring Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**clear-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** monitoring command. Therefore when you restart the router, the effects of this command are erased.

**Syntax:**

**disable** group . . .
port

**group** *group#*
Suppresses transfer of SDLC Relay frames to or from a specific group (group#).

**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** monitoring command. Therefore when you restart the router, the effects of this command are erased.

**Syntax:**

**enable** group . . .
port

**group** *group#*
Allows transfer of SDLC Relay frames to or from the specified group (group#).

**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:**

**list** all
group . . .

**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).

**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.

**Packets (fwrd and disc)**
Indicates how many packets were forwarded (fwrd) and discarded (disc) for that port.

**IP Address**
Indicates the IP address of the remote port.

**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
```

## SDLC Relay Interfaces and the GWCON Interface Command

While SDLC Relay interfaces have their own monitoring 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 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands.)

# Chapter 47. Using SDLC Interfaces

This chapter how to use the SDLC interface and includes the following sections:

- "Basic Configuration Procedure"
- "SDLC Configuration Requirements" on page 590
- "Configuring Switched SDLC Call-In Interfaces"

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.

## 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.

## Configuring Switched SDLC Call-In Interfaces

A switched SDLC call-in interface allows a PU type 2.0 device to dial into a 2216 using a switched SDLC line, providing an additional connectivity option to your network. The interface is restricted to PU type 2.0 devices and can run DLSw only.

**Note:** You cannot configure APPN over a switched SDLC call-in interface.

To configure a switched SDLC call-in interface:

1. Configure a V.25bis base network:

```
Config> set data-link v25bis 2
Config> net 2
V25bis Config>
 (configure the V25bis net)
```

   See "Chapter 49. Using the V.25bis Network Interface" on page 613 for more information about configuring V25bis.

   **Note:** Any physical layer parameters such as the **encoding type** and **full** vs. **half duplex** are configured on the V.25bis interface and not on the Switched SDLC dial circuit interface.

2. Add a dial circuit device:

```
Config> add device dial
```

3. Set the data link for the dial circuit interface to SDLC. In this example, the dial circuit is interface 3.

```
Config> set data-link sdlc 3
```

4. Configure the dial circuit:

```
Config> net 3
Dial circuit config> set net 2 1
Dial circuit config> encapsulator
sdlc config>
   (configure SDLC)
sdlc config> exit
Dial circuit config> exit
Config>
```

5. Configure DLSw:

```
Config> prot dls
DLSw protocol user configuration
DLSw config> add sdlc
Interface # [0]? 3
SDLC Address or 'sw' (switched dial-in) [sw]? sw 2
Source MAC address [4000112402C1]? 4000003174d2
Source SAP in hex [4]?
Destination MAC address [000000000000]? 400000000004 3
Destination SAP in hex [0]? 4 4

XID0 block num in hex (0-0xfff) [0]? 017
XID0 id num in hex (0-0xfffff) [0]? 00001
For a switched dial-in link station .....
  - PU type is forced to be 2
  - Configured XID block/id num is used to override
     fields in the XID0 from the SDLC station
      - if block/id set to zeroes, XID0 is not modified
      - otherwise configured fields are put into XID0
  - Poll type is not configured (not used)
DLSw config>li sdlc all
Net Addr   Status    Source SAP/MAC   Dest SAP/MAC      PU  Blk/IdNum  PollFrame
 3  FF(sw) Enabled   04 4000003174D2  04 400000000004   2   017/00001  TEST

DSLw config> exit
Config>
```

1 You will not be able to set any other dial circuit parameters as the software will take defaults for all other parameter values. For information about the defaults, see “Encapsulator” on page 655.

2 Specifying “sw” indicates that this is a switched SDLC call-in interface.

3 The destination MAC address cannot be all 0s. If you specify or default to a value of 0, the software will prompt you for a valid address.

4 The destination SAP cannot be 0. If you specify or default to a value of 0, the software will prompt you for a valid address.

See the “Using and Configuring DLSw” and the “Monitoring DLSw” chapters of *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1* for additional information about configuring DLSw.

## 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 configuration and an APPN configuration exist for the same SDLC interface, the first protocol to come active will own the SDLC interface.

# Chapter 48. Configuring and Monitoring SDLC Interfaces

This chapter describes the SDLC configuration and operational commands.

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 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" on page 3 for information related to the configuration environment.

# 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 83 lists SDLC configuration commands and their function.

*Table 83. SDLC Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

# 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:**

**add** 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:**

**delete** station *name* or *address*

## Disable

Use the **disable** command to prevent connections from being created with a SDLC link station.

**Syntax:**

**disable** link
station . . .

**link** Prevents the transmitting and receiving of data to all configured SDLC link stations on the interface.

**station** *name* **or** *address*
Prevents the transmitting and receiving of data to the specified end station (station name or address).

## Enable

Use the **enable** command to enable connections to remote SDLC link stations.

**Syntax:**

**enable** link
station

**link** Allows subsystems in the router (for example, DLSw) to use SDLC's facilities.

**station** *name* **or** *address*
Allows connections to the specified secondary remote end station (link station name).

## List

Use the **list** command to display configuration information on one or all SDLC link stations.

**Syntax:**

**list** link
station *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
            Leading flags:  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.

**Leading Flags**
The number of flags sent if the interframe delay is not sufficient for a response to the device on the other end of this link.

**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.

**Note:** Physical layer parameters such as **duplex type**, **speed**, **cable type**, **encoding**, **clocking**, **leading flags**, and **inter-frame delay** do not apply for SDLC dial circuit interfaces and are not displayed by the **list link** command.

**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:**

**set**
- link cable*
- link clocking*
- link duplex* . . .
- link encoding* . . .
- link frame-size
- link group poll* ...

link idle* . . .

link inactivity ...

link inter-frame delay*

link leading flags*

link modulo . . .

link name

link poll . . .

link role* . . .

link snrm

link speed*

link type* . . .

link xid/test

station *address* . . .

***Note:** These commands are not available for SDLC dial circuit interfaces.

**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 84 lists the cable types you can configure on the various adapters.

*Table 84. Cable types for 2216 Interfaces*

| Adapter Type | Cable Iype |
|---|---|
| 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.

**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 86 on page 600 to determine the clock speeds you can set for the various adapters when internal clocking is used.

**link duplex** *full* **or** *half*

Configures the SDLC line for *full-duplex* or *half-duplex* signalling. *Half-duplex* means that the 2210/2216 raises RTS and expects to see CTS before it will transmit data. *Full-duplex* means that the 2210/2216 does not wait for CTS to be raised before is transmits data.

**Note:** The duplex type does not control how SDLC operates at the SDLC protocol level. The 2216/2210 only supports two-way alternating mode which is sometimes also referred to as SDLC half-duplex.

**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.

**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 85.

*Table 85. Valid Values for Frame Size in Link Frame-Size Command*

| Minimum | Maximum | Default |
|---|---|---|
| 128 | 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 - 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.

**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.

**Note:** If you configure non-zero inter-frame delay for a SDLC interface on the 8-port EIA- 232E adapter, 6-port V.35/V.36 adapter, or 8-port X.21 adapter, you must configure the line speed using the **set link speed** command.

**Example:**

```
set link inter-frame delay
Transmit Delay Counter [0]?
```

**link leading flags**

Sets the number of leading flags. Use this command when the inter-frame delay is not sufficient to allow delay a response from the 2216 to another device. This command should also be used to set a leading flags delay if you are using half-duplex modems that are not capable of receiving a packet as soon as the modem raises the CTS modem signal.

**Valid values:** 0 to 100

**Default value:** 0

**Example:**

```
set link leading flags
Leading flags delay [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.

**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 [2.0]?
```

**link role** *primary* **or** *secondary* **or** *negotiable*

Configures the interface as an SDLC primary, secondary, or negotiable link station (default is primary).

**Notes:**

1. For DLSw, ***negotiable*** uses X'FF' (broadcast address) for the initial poll.

   When using broadcast address to negotiate the role, the link uses a default SDLC configuration.

   When ***primary*** is the link role, the link performs an initial poll to a specific address.
2. For APPN point-to-point or negotiable, the broadcast address is used for the initial poll. For primary multipoint, the specific address is used.
3. For switched SDLC, the device must be primary, so **link role type** is not configurable for SDLC dial circuit interfaces.

**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. The range is from 2400 to 2048000 bps. Use Table 87 to determine the clock speeds you can set for the various adapters.

*Table 86. 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 87 for line speeds supported when external clocking is used.

*Table 87. 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.

**Note:** For switched SDLC, the link is always point-to-point, so **link type** is not configurable for SDLC dial circuit interfaces.

**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.

**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]?
```

## 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” on page 3 for information related to the monitoring environment.

## SDLC Monitoring Commands

This section summarizes and then explains the SDLC console and related commands. Use these commands to gather information from the database. Table 88 lists SDLC monitoring commands and their function.

*Table 88. SDLC Monitoring Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**add** station

For an example and for additional information on the **add** command, see “Add” on page 592 .

## 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: clear** link

station ...

**link** *name* **or** *address*
Clears the counters for an SDLC interface.

**station** *name* **or** *address* **or all**
Clears counters for a specific station or for all stations.

## 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:**

**delete** station *name* or *address*

## 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: disable**
link

station . . .

**link** Prevents connection on all configured SDLC link stations on the interface by terminating all connections.

**station** *name* **or** *address*
Prevents connection to the specified end station (link station name) by terminating any existing connection.

## Enable

Use the **enable** command to enable connection establishment with remote SDLC link stations without affecting the SDLC configuration SRAM.

**Syntax:**

**enable** link

station . . .

**link** Allows subsystems (for example, DLSw) to use SDLC's facilities.

**station** *name* **or** *address*
Allows connections to the specified end station.

## List

Use the **list** command to display statistics specific to the data link layer and the interface.

**Syntax:**

**list** link configuration

link counters

station . . .

**link configuration**
Displays information for all configured SDLC link stations on the interface.

For an example and for additional information on the **list** command, see "List" on page 594.

**link counters** Displays information for the SDLC counters since the last router restart or the last clear counters.

**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.

**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.

**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 Acknowledgment 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.

**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
```

**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
```

**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
```

## 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 type* . . .

link xid/test

station . . .

***Note:** These commands are not supported on SDLC dial circuit interfaces.

**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.

**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.

**Notes:**

1. For DLSw, ***negotiable*** uses X'FF' (broadcast address) for the initial poll.
   When using broadcast address to negotiate the role, the link uses a default SDLC configuration.
   When ***primary*** is the link role, the link performs an initial poll to a specific address.
2. For APPN point-to-point or negotiable, the broadcast address is used for the initial poll. For primary multipoint, the specific address is used.
3. For switched SDLC, the device must be primary, so **link role type** is not configurable for SDLC dial circuit interfaces.

**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 Acknowledgment (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.

**Note:** For switched SDLC, the link is always point-to-point, so **link type** is not configurable for SDLC dial circuit interfaces.

**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.

**Note:** Examples for, and explanations of, the following parameters can be found in the SDLC configuration chapter at “Set” on page 596 .

**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:**

**test** station *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.

# SDLC Interfaces and the GWCON Interface Command

While the SDLC interface has a console process for operational 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 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands” on page 99.)

# 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 49. Using the V.25bis Network Interface

The V.25bis interface allows routers to establish serial connections over switched telephone lines using V.25bis modems. This chapter describes how to use the V.25bis interface. It includes the following sections:

- "Before You Begin"
- "Configuration Procedures"

**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.

## Before You Begin

Before you configure V.25bis on the router, make sure you have the following:

- V.25bis modems that support synchronous V.25bis commands and the 1988 ITU/CCITT V.25bis specification.
- If your modem does not automatically detect answer originate, you must:
  - 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.25bis. The tasks you need to perform are:

1. Adding V.25bis addresses
2. Configuring V.25bis parameters
3. Adding dial circuits
4. Configuring dial circuits

**Note:** You must restart the router for changes to the V.25bis configuration to take effect.

## Adding V.25bis Addresses

You need to add a V.25bis address for each local V.25bis interface as well as for each destination. The V.25bis 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 30 characters that are in the valid format of the connected V.25bis modem. For additional information consult your modem manual.

  **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 (dependent 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.25bis 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 [1-30 digits][]? 19095551234
```

# Configuring the V.25bis Interface

This section explains how to configure the V.25bis interface. To configure, do the following:

1. To set up a serial line interface for V.25bis, 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.25bis Parameters

The following are optional V.25bis parameters you can set. For a complete description of these commands, see “V.25bis Configuration Commands” on page 617 .

- 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.
- The **set duplex** command specifies the duplexing mode for the call.
- The **set encoding** command sets the encoding for the call.
- 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.25bis 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). You can also set the dial circuit to use Frame Relay (FR) or SDLC.

## Configuring Dial Circuits

This section describes how to configure a dial circuit. For a complete description of the dial circuit commands, see “Chapter 53. Using Dial Circuits” on page 653.

**Note:** If the encapsulator type is SDLC, the only dial circuit parameter that you can set is the base net number.

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.25bis interface. The Base net is the V.25bis 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 or WAN Reroute 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. Set the duplexing mode for the circuit using the **set duplex** command.
9. Set the encoding mode for the circuit using the **set encoding** 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
    ```

# Chapter 50. Configuring and Monitoring the V.25bis Network Interface

This chapter describes the V.25bis configuration and operational commands and GWCON commands. It includes the following sections:



## Accessing the Interface Configuration Process

Use the following procedure to access the V.25bis configuration process.

1. At the OPCON prompt, enter the **talk** command and the PID for CONFIG. (For more detail on this command, refer to Chapter 4. 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.
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.
4. 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.25bis configuration prompt now displays on the console.

## V.25bis Configuration Commands

Table 89 summarizes and the rest of the section explains the V.25bis configuration commands. These commands allow you to display, create, or modify a V.25bis configuration. Enter the V.25bis configuration commands at the `V.25bis Config>` prompt.

*Table 89. V.25bis Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| List | Displays the V.25bis configuration. |
| Set | Sets the local address, connect, disconnect, and no answer timeouts, number of retries after no answer, the duplexing mode, command delay timeout, and encoding. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

# List

Use the **list** command to display the current V.25bis configuration.

**Syntax:**

**list**

**Example:**

```
list
          V.25bis Configuration

Duplex                         =  Full
Encoding                       =  NRZ
Local Network Address Name     = v403
Local Network Address          = 15088982403

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
```

**Duplex**
Displays the duplex mode for the interface once the dial connection has been established.

**Encoding**
Displays the transmission encoding scheme for the interface once the dial connection has been established. Encoding is either NRZ (non-return to zero) or NRZI (non-return to zero inverted).

**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:**

**set** command-delay timeout . . .
connect-timeout . . .
disconnect-timeout . . .
duplex
hdlc cable . . .
hdlc encoding . . .
hdlc speed . . .
local-address . . .
retries-no-answer . . .
timeout-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.

**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.

**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.

**duplex**

Specifies the duplex type of the line.

When full-duplex is configured, the RTS modem signal remains asserted once the dial connection has been established.

When half-duplex is configured, the router raises RTS when it is time to transmit and waits for CTS to be asserted by the modem. After CTS is asserted, the router transmits data packets and then drops RTS when the router is through transmitting to let the peer device respond.

Only configure half-duplex when using the V.25bis interface to handle switched SDLC and the attached modem requires the half-duplex mode of operation.

**Notes:**

1. Duplex must be full for PPP or Frame Relay circuits.
2. If you configure half-duplex for a V.25bis interface on the 8-port EIA-232E adapter, you must configure the line speed using the **set hdlc speed** command.

**Valid values:** full or half

**Default value:** full

**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.

**hdlc encoding**

Sets the HDLC transmission encoding scheme as NRZ (non-return to zero) or NRZI (non-return to zero inverted). Most configurations use NRZ. The configured encoding is used for the end-to-end connection.

**Note:** Although you might configure NRZI, the exchange between the DTE and the modem (as described by CCITT recommendation, *V.25bis*) uses NRZ as the encoding scheme.

**Valid values:** NRZ or NRZI

**Default value:** NRZ

**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.25bis interface.

**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.

**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.

## Accessing the Interface Monitoring Process

To access the interface monitoring process for V.25bis, enter the following command at the GWCON (+) prompt:

```
+ network #
```

Where *#* is the number of the V.25bis serial line. You cannot directly access the V.25bis monitoring process for dial circuits, but you can monitor the dial circuits that are mapped to the serial line interface.

**Note:** V.25bis interfaces also have ELS troubleshooting messages that you can use to monitor V.25bis related activity. See the *IBM Nways Event Logging System Messages Guide* for further details.

## V.25bis Monitoring Commands

This section summarizes and explains the V.25bis operating commands. These commands allow you to view the calls, circuits, parameters, and statistics of the V.25bis interfaces.

Enter the V.25bis monitoring commands at the `V.25bis>` prompt.

*Table 90. V.25bis Monitoring Command Summary*

| Monitoring Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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. |

*Table 90. V.25bis Monitoring Command Summary (continued)*

| Monitoring Command | Function |
|---|---|
| Circuits | Shows the status of all data circuits configured on the V.25bis interface. |
| Parameters | Displays the current parameters for the V.25bis interface. (This command is similar to the `V.25bis Config>` list command.) |
| Statistics | Displays the current statistics for the V.25bis interface. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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
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.25bis port.

**Syntax:**

**circuits**

**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 monitoring 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.25bis serial line configuration. Note that this is the same information displayed in the `V.25bis Config>` list command.

**Syntax:**

**parameters**

**Example:**

```
parameters
         V.25bis port Parameters

Local Network Address Name    = v402
Local Network Address         = 15088982402

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.25bis interface.

**Syntax:**

**statistics**

**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.

## V.25bis and the GWCON Commands

While V.25bis has its own monitoring 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.25bis serial interface causes the current call to be dropped and re-dialed.

For more information on the GWCON command, see “Chapter 10. The Operating/Monitoring Process (GWCON - Talk 5) and Commands” on page 99.

## Statistics for V.25bis Interfaces and Dial Circuits

Use the **interface** command at the GWCON (+) prompt to display statistics for V.25bis serial line interfaces and dial circuits.

To display the following statistics for a V.25bis serial line interface, use the **interface** command followed by the *interface number* of the V.25bis 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 list 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.25bis 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 51. Using the ISDN Interface

**Important**

The original 2216 ISDN T1 and E1 Single Port LICs (IBM PN 11J7466 or 78H6147 and 11J7465 or 78H6148 respectively), do not support MAC address assignment for DIALs clients. This assignment is only necessary for customers who want to use the NetBIOS and/or SNA protocols natively in the DIALs Remote LAN Access environment. All other ISDN functions and DIALs functions (for example Dial-in IP and IPX) will work correctly without an upgrade. The newer ISDN T1 and E1 Single Port LICs support MAC address assignment for DIALs clients and all DIALs functions. Customers with the original ISDN LICs who want to use NetBIOS and/or SNA DIALs functions should contact their IBM service representative to upgrade to the newer LICs.

This chapter describes the Integrated Services Digital Network (ISDN) interface on the IBM 2216. It includes the following sections:



## 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.

The I.430, I.431 and Channelized T1/E1 are not switched. They are permanent leased line type connections.

## ISDN Adapters and Interfaces

The IBM 2216 supports the following ISDN-PRI adapters:

- 1-Port Channelized E1 ISDN-PRI
- 1-Port Channelized T1/J1 ISDN-PRI
- 4-Port Channelized E1 ISDN-PRI
- 4-Port Channelized T1/J1 ISDN-PRI

The PRI/Channelized adapters have an integrated CSU/DSU, so an external CSU/DSU is not required.

The interfaces are:

**Note:** If you are upgrading from BRI to PRI from talk 6, you must clear the ISDN and dial configurations first, then bring up PRI and configure for PRI.

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 thirty 64-Kbps (kilobits per second) bearer (B) channels and one 64-Kbps data (D) channel. The B channels are used as HDLC frame delimited 64-Kbps 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 twenty-three 64-Kbps B channels and one 64-Kbps D channel.
  - E1 supports thirty 64-Kbps B channels and one 64-Kbps 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

There are four types of dial circuits:

- Static circuits (or link)

  **Notes:**

  1. I.430, I.431 and Channelized T1/E1 are leased line connections and therefore do not dial.
  2. ISDN considers X.25 traffic over the D-Channel as a static circuit. However, you could configure the X.25 circuit as a PVC or SVC using the **encapsulator** command.
- 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
- Dial-in circuits are used to provide remote clients access to resources on the network.

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), Frame Relay or X.25 for D-Channels only. These protocols are configured to operate over the dial circuits.

**Note:** You can assign a destination name to a **connection list** (add ISDN address) 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.

Routable protocols and bridging and routing features 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 switch, 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, the local network address is obtained from the physical interface configuration and you set a destination addresses for the circuit.

## Circuit Contention

An ISDN PRI T1/J1 interface can support a maximum of 23 active calls, and an ISDN PRI E1 interface can support a maximum of 30 active calls. There can be more dial circuits configured on an ISDN interface than active calls supported. If a dial circuit attempts a call when the ISDN interface has all calls active, there are two possibilities: 1) If the dial circuit has a higher priority than a dial circuit with an active call, the active call will be terminated for the low priority dial circuit and a call will be attempted for the low priority dial circuit and a call will be attempted for the higher priority dial circuit. 2) If the dial circuit does not have a higher priority than any dial circuits with active calls, no call will be made. The router will drop packets sent by protocols on dial circuits that cannot connect to their ISDN destination.

**Note:** There is no circuit contention when you are running X.25 over the D-channel because the D-Channel is always available for the X.25 connection.

See “Set” on page 657 for more information about priority.

## Cost Control Over Demand Circuits

Dial-on-demand circuits always appear to be in the Up state to the 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 static routes and services and disable the routing protocols (RIP, SAP) over the dial circuits. Another option is to configure low-frequency RIP and SAP update intervals, although this does not prevent RIP and SAP from broadcasting routing information changes as they occur. You should also enable IPX Keepalive filtering, which prevents keepalive and serialization packets from continually activating the dial-on-demand link.

## Call Verification

This ISDN implementation uses a proprietary line 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 is receiving the call. The line 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 line ID message, the call may be rejected. The line ID exchanges occur on the B channel.

When connecting to routes that do not support logical ids (LIDS), you can suppress the lid exchange using the config option under the individual dial circuit.

```
config> set lid_used
```

On the incoming side, if this variable is set, the call is transferred to the first dial circuit configured for any inbound or with the caller's phone number in the inbound destination field.

## ISDN Cause Codes

This ISDN implementation specifies a cause code that will stop the router from attempting 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 application will not attempt further connection processing through this interface.

Cause code interpretations:

1. If cause0 is not ″0x5″ ignore the cause code.
2. If cause0 is ″0x5″ look at cause1. If the high-order (most significant) bit of cause1 is `ON`, set it to `OFF`.
3. Convert the result to decimal and look up the meaning in the following table, which is taken from *ITU-T Recommendation Q.850*.

*Table 91. ISDN Q.931 Cause Codes*

| Code | Cause |
|---|---|
| 1 | Unallocated (unassigned number) |
| 2 | No route to specified transit network |

*Table 91. ISDN Q.931 Cause Codes (continued)*

| Code | Cause |
|---|---|
| 3 | No route to destination |
| 6 | Channel unacceptable |
| 7 | Call awarded and is being delivered in an established channel |
| 16 | Normal call clearing |
| 17 | User busy |
| 18 | No user responding |
| 19 | No answer from user (user alerted) |
| 21 | Call rejected |
| 22 | Number changed |
| 26 | Non-selected user clearing |
| 27 | Destination out of order |
| 28 | Invalid number format (address incomplete) |
| 29 | Facility rejected |
| 30 | Response to STATUS ENQUIRY |
| 31 | Normal, unspecified |
| 34 | No circuit/channel available |
| 38 | Network out of order |
| 41 | Temporary Failure |
| 42 | Switching equipment congestion |
| 43 | Access information discarded |
| 44 | Requested circuit/channel not available |
| 47 | Resource unavailable, unspecified |
| 49 | Quality of Service not available |
| 50 | Requested facility not subscribed |
| 57 | Bearer capability not authorized |
| 58 | Bearer capability not presently available |
| 63 | Service or option not available, unspecified |
| 65 | Bearer capability not implemented |
| 66 | Channel type not implemented |
| 69 | Requested facility not implemented |
| 70 | Only restricted digital information bearer capability is available |
| 79 | Service or option not implemented, unspecified |
| 81 | Invalid call reference value |
| 82 | Identified channel does not exist |
| 83 | A suspended call exists, but this call identity does not |
| 84 | Call identity in use |
| 85 | No call suspended |

*Table 91. ISDN Q.931 Cause Codes (continued)*

| Code | Cause |
|---|---|
| 86 | Call having the requested call identity has been cleared |
| 88 | Incompatible destination |
| 91 | Invalid transit network selection |
| 95 | Invalid message, unspecified |
| 96 | Mandatory information element is missing |
| 97 | Message type nonexistent or not implemented |
| 98 | Message not compatible with call state or message type nonexistent or not implemented |
| 99 | Information element nonexistent or not implemented |
| 100 | Invalid information element contents |
| 101 | Message not compatible with call state |
| 102 | Recovery on timer expiry |
| 111 | Protocol error, unspecified |
| 127 | Interworking, unspecified |

# Sample ISDN Configurations

The following topics show several typical ISDN configurations.

## Frame Relay over ISDN Configuration

Figure 47 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).

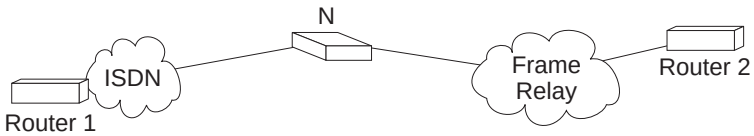



*Figure 47. Frame Relay over ISDN Configuration*

**Note:** N could be either an ISDN TA connected to the FR switch, or an ISDN card in a FR switch.

## WAN Restoral Configuration

Figure 48 on page 635 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 61. Using WAN Restoral” on page 739.

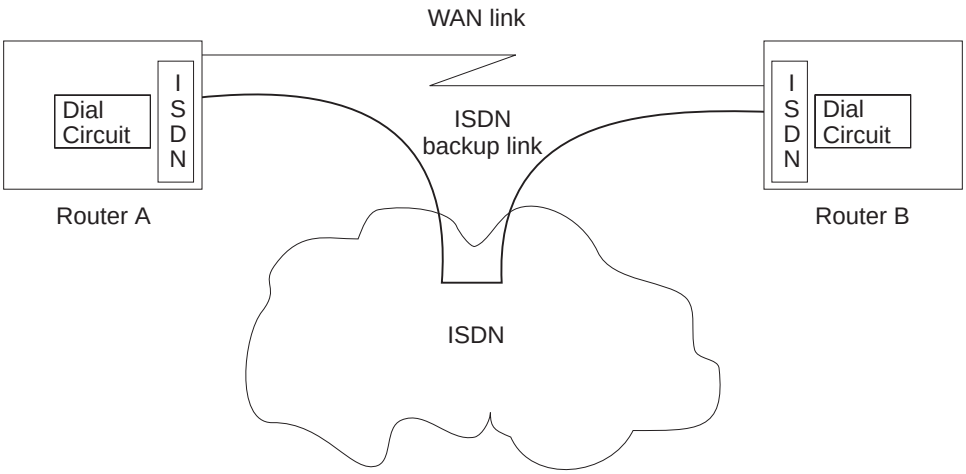



*Figure 48. Using ISDN for WAN Restoral*

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.

## Channelized T1/E1

When configured for channelized, the Channelized/PRI adapter allows you to get Fractional/Channelized T1/J1/E1 support. You can have channels of 56-Kbps or N*64-Kbps. This will let you multiplex multiple leased lines connections (for example: using V.35 at 56-Kbps) into one physical connection.

To configure a T1 or E1 Primary adapter as channelized:

1. Select “Channelized” as the switch variant for the ISDN interface.
2. Configure the time slots to be used for this ISDN interface when you configure the dial circuit. See “Set” on page 657 for more information.

**Example of configuring a Channelized T1 interface:**

```
Config>n 6
ISDN Config>set switch chan
ISDN Config>list


                  ISDN Configuration

Maximum frame size in bytes      = 2048
Switch Variant/Service Type      = Channelized
Available Timeslots: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24


Config>n 7
Circuit config:  7>set net 6
Circuit config:  7>set timeslot 2 3 4 24
Circuit config:  7>list

Base net                      = 6
Idle character                = 7E
Bandwidth                     = 64 Kbps
Timeslot                      = 2 3 4 24
```

**Note:** If this were an E1 circuit, the available timeslots would be 1 to 31.

# Requirements and Restrictions for ISDN Interfaces

## Switches/Services Supported

The ISDN Primary Rate Interface (PRI) supports the following switches/services:

| 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 |
| Native I.431 | I431 (See “I.431 Switch Variant” on page 641 .) |
| Channelized T1/E1 | CHANNELIZED |

## 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 D channel.

## 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/Services Supported” for the list of switches.

**Note:** Additional parameters, such as TEI and SPID may be required based on your Switch Type and your service provider.

## 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 643 for more information about configuration commands.

1. Specify the type of switch/service 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/Services Supported” on page 636 for the list of switches/services. For example:

```
ISDN Config>set switch net5
```

This is the software type running at the switch (for example, DMS100 means running DMS100 Custom software).

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
   ```

   **Note:** This is what we will send in the Calling Party Number field of the ISDN Setup message.

3. Set the directory number of the local port.
4. To set the frame size, use the **set framesize** command. For example:

   ```
   ISDN Config>set framesize
   Framesize in bytes (1024/2048/4096/8192) [1024]? 2048
   ```

   **Note:** If you choose a frame size of 1024, PPP will not work over the ISDN dial circuit, since the minimum frame size for PPP is 1500.

   For more information about setting the ISDN framesize, see “Set” on page 644.

### 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 643.

- For all ISDN switches except INSPRI, you can configure the limit for the number of calls to an address. 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.

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 PRI, you need to configure T1/J1 or E1 for each adapter, depending upon the adapter.″

### 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.

**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, SDLC, and SRLY) are not supported over ISDN.

## 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
```

   **Note:** If the dial circuit data link type is X.25 or the base net switch variant is I.43x or channelized, the following steps (3-10) do not apply.

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. 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
```

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). The receiving router must have an inbound destination address that matches the lid_out_address on the sending router in order for the dial circuits to connect. 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 an outbound call 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.

   **Note:** Only outbound dial-on-demand circuits will be brought down.

See “Set” on page 657 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. This command is for inbound circuits only. 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
```

## I.431 Switch Variant

The I.431 switch variant should be configured when running a leased line over ISDN PRI.

## Native I.431 Support

When configuring for Native I.431 support, only one dial circuit should be used. It should be attached to the base net. The I.431 only runs on the ISDN PRI T1 adapter. The speed is fixed at 1.5 Mbps.

**Example: Base ISDN net**

```
Config>n 5
ISDN Config>set sw i431
ISDN Config>list all
                  ISDN Configuration

Maximum frame size in bytes      = 2048
Switch Variant                   = I431 PRI
```

**Example: Dial Circuit**

```
Config>n 6
Circuit config:   6>set net 5
Circuit config:   6>list all


Base net                       = 5
```

# Chapter 52. Configuring and Monitoring the ISDN Interface

This chapter describes the ISDN commands and GWCON commands. It includes the following sections:



**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*

## ISDN Configuration Commands

Table 92 describes the ISDN configuration commands, and the following sections explain the commands. Enter these commands at the `ISDN Config>` prompt.

*Table 92. ISDN Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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 DN0 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, SPIDS, and TEI. |
| Cause Codes | Stops further processing attempts to establish a connection through an interface. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## List

The **list** command displays the current ISDN configuration.

**Syntax:**

**list** 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    =  180  Retries =   2
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 DN0 entries you set using the **set DN0 entry** command.

**Syntax:**

| | |
|---|---|
| **remove** | DN0-entry... |

**Example:**

```
remove DN0
```

## 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... |
| | frame-type[1] |
| | interface |
| | local-address-name... |
| | RAI-type[1] |
| | retries-call-address... |
| | switch-variant... |
| | dn0... |

## Cause Codes

Use the **Cause Code** command to prevent the router from retrying to establish a connection through the ISDN interface when it receives a “specified” (valid value) response. Enter these commands at the `Cause Config>` prompt.

**Syntax:**

| | |
|---|---|
| **cause** | ? (Help) |
| | add |
| | list |
| | remove |
| | exit |

1. Channelized only

*Table 93. ISDN Cause Codes Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add | Adds cause code entries to the ISDN configuration. |
| List | Displays the cause code lists for the ISDN configuration. |
| Remove | Removes cause code entries from the ISDN configuration. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

**Add** Use the **add** command to add a cause code to an ISDN configuration.

**Valid Values:** Any hexadecimal value between 01 and FF

**Default Value:** None

**Syntax:** cause code add *value*

**Example:** **`add FF`**

**Remove**

Use the **remove** command to remove a cause code from an ISDN configuration.

**Valid Values:** Any hexadecimal value between 01 and FF

**Default Value:** None

**Syntax:** cause code remove *value*

**Example:** **`remove FF`**

**List** Use the **list** command to show the cause code list of an ISDN configuration.

**Syntax:** cause code list

## Accessing the Interface Monitoring Process

To access the interface monitoring 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 monitoring process for dial circuits, but you can monitor the dial circuits that are mapped to the ISDN interface.

# ISDN Monitoring Commands

The following sections explain the ISDN operating 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 94. ISDN Monitoring Command Summary*

| Monitoring Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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 all available channels are in use, if the maximum retries are used up and the router is waiting for the timer to count down, or if layer 1 is up, but layer 2 is down.

**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.

**Syntax:**

**channels**

## 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:**

**circuits**

**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 monitoring 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 0:               Automatic
TEI 1:               Automatic
X.25 TEI:               21
Switch Variant:      AT&T 5ESS (United States)
Multipoint Selection:    Multipoint
Directory Number 0:      20
Outbound call address Timeout:     180       Retries:     0
```

## 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 using I.431:**

```
statistics
Transmit                   Receive

Packets              0     Packets             0
Bytes                0     Bytes               0
Overflow         68480     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:    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
```

## ISDN and the GWCON Commands

While ISDN has its own monitoring 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 102 for a sample output from the **configuration** command.

# Chapter 53. Using Dial Circuits

This chapter describes how to use dial circuits on a dial circuit interface mapped to a V.25bis or ISDN interface.

**Notes:**

1. PPP dial circuit interfaces can use an ISDN, or a V.25bis network as the base network interface.
2. FR dial circuit interfaces can use an ISDN or a V.25bis network as the base network interface.
3. Switched SDLC Call-In dial circuit interfaces use a V.25bis network as the base network interface.
4. X.25 circuits can be used over ISDN D-Channel for BRI.

For information on how to configure dial circuits for use with:

- ISDN interfaces, see “Chapter 51. Using the ISDN Interface” on page 629.
- V.25bis interfaces, see “Chapter 49. Using the V.25bis Network Interface” on page 613 .

# Chapter 54. Configuring Dial Circuits

This section describes the dial circuit configuration and operational commands.

## Dial Circuit Configuration Commands

"Chapter 53. Using Dial Circuits" on page 653 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 95. Dial Circuit Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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, lid_out address, inbound destination, and self-test delay. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## 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.

## Encapsulator

Use the encapsulator command to enter the configuration process for the link-layer protocol (for example. PPP, Frame Relay, X.25 , SDLC) that is running on the dial circuit interface.

**Note:** The default for a dial circuit interface created via the **add device dial-circuit** command is PPP. If you want to change the link layer type to Frame Relay, use the **set data-link frame-relay** command at the `Config>` prompt. If you want to change the link layer type to SDLC, use the **set data-link sdlc** command at the `Config>` prompt.

**Syntax:**

**encapsulator**

The following example shows that the PPP configuration process is entered when the encapsulator command is used for a PPP dial circuit interface.

**Example:**

```
encapsulator
Point-to-Point user configuration
PPP Config>
```

Be aware of the following when you configure a dial circuit that uses a V.25bis interface as the base network:

- The V.25bis interface pre-defines clocking as external. The modem (DCE) controls the clock speed. You cannot configure clocking, encoding, and other HDLC parameters as part of the 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 34. Configuring Serial Line Interfaces” on page 387 or refer to “Chapter 41. Using Point-to-Point Protocol Interfaces” on page 511.

For information on configuring the Frame Relay protocol, see “Chapter 39. Using Frame Relay Interfaces” on page 457 or “Chapter 40. Configuring and Monitoring Frame Relay Interfaces” on page 475.

For information on configuring or monitoring SDLC interfaces, see “Chapter 47. Using SDLC Interfaces” on page 589 or “Chapter 48. Configuring and Monitoring SDLC Interfaces” on page 591.

For information on configuring or monitoring X.25 interface, see “Chapter 36. Configuring and Monitoring the X.25 Network Interface” on page 397.

To return to the `Circuit Config>` prompt, use the **exit** command.

## List

Use the **list** command to display the current dial circuit configuration.

For more information about I.430 and I.431, see “I.431 Switch Variant” on page 641.

**Syntax:**

**list**

**Example:**

```
list
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 LID 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. This is an alternate comparison address name used by the LID 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 (for example: ISDN or V.25bis), configure the dial circuit for inbound and/or outbound calls, and set destination addresses, inbound addresses, idle timeout, and self-test delay.

**Note:** If you are running SDLC, I.430, I.431, Channelized, or X.25 on a dial circuit, you will be unable to use the **set** command to change the following parameters as the software will use specific defaults:

- Calls - inbound
- Destination - default address
- Inbound destination - no destination inbound address
- Any_inbound - any_inbound is set
- Idle - 0
- Lid_out_addr - no LID name
- Lid_used - disabled
- Priority - 8
- Self_test_delay

**Syntax:**

**set** any_inbound

bandwidth

calls...

destination...

inbound destination...

idle...

idle-char

lid_out_addr...

lid_used

net...

priority...

selftest-delay...

timeslot . . .

**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.

**bandwidth** *kbps*
Sets the bandwidth, in Kbps, for I.430 and Channelized T1/E1 circuits.

**Valid values:**

For I.430: 64 or 128

For Channelized: 56 or 64

**Default value:** 64

**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.

**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 LID 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 LID 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`

**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.

**Notes:**

1. For WAN Restoral operations, you must set the idle timeout to 0.
2. On a I.43x, X.25 or Channelized circuit, you cannot set this parameter.

**idle-char**

Specifies the idle character used for I.43x or channelized circuits.

**Note:** You cannot configure this parameter for regular ISDN circuits.

**Valid values:** 7E or FF

**Default value:** 7E

**Example:** `set idle-char 7E`

**lid_out_addr** *address_name*

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.

**lid_used [enabled or disabled]**

Suppresses the exchange of logical ids for circuits to devices that do not support logical ids.

**Valid values:** Enabled or disabled

**Default value:** Disabled

**net** *#* Sets the base circuit number to *#* of serial line interface to which you want to map this circuit.

**Example:**

```
Circuit Config> set net
Base net for this circuit [ ]? 2
```

**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.25bis 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.

**timeslot** *list of slots*

Specifies a slot or list of slots to use for this dial circuit. Your service provider will issue the number of the slots you can use for the circuit. Specify the list as slot numbers separated by blanks.

**Note:** You can only use this parameter for Channelized T1/E1 circuits.

**Valid values:**

For Channelized T1: 1 to 24

For Channelized E1: 1 to 31

**Default value:** None

**Example:** `set timeslot 1 4 5 8`

# Chapter 55. Using Layer 2 Tunneling Protocol (L2TP)

Layer Two Tunneling Protocol (L2TP) is a standards track IETF proposed standard protocol for tunneling of PPP across a packet oriented data network such as UDP/IP. L2TP is connection oriented.

## Overview of L2TP

L2TP allows many separate and autonomous protocol domains to share a common access infrastructure including modems, Access Servers, and ISDN routers. L2TP permits the tunneling of the PPP link layer, for example, HDLC and asynchronous HDLC. Using these tunnels, it is possible to disassociate the location of the contacted dial-up server from the location that provides access to the network.

Traditionally, dial-up network service on the Internet is provided for registered IP addresses only. L2TP defines a new class of virtual dial-up application that allows multiple protocols and unregistered IP addresses on the Internet. This class of network application is useful for supporting privately addressed IP, IPX, and AppleTalk dial-ups through PPP across an existing Internet infrastructure.

The support of these multiprotocol virtual dial-up applications is beneficial to end users, enterprises, and Internet service providers because it allows the sharing of significant investments in access and core infrastructure and allows end users to use local calls when accessing the services.

L2TP also enables the secure use of existing investments in non-IP protocol applications within the existing Internet infrastructure.

Figure 49 shows a sample L2TP network using ISDN. The network could use any media type between the L2TP Network Access Concentrator (LAC) and the L2TP Network Server (LNS).

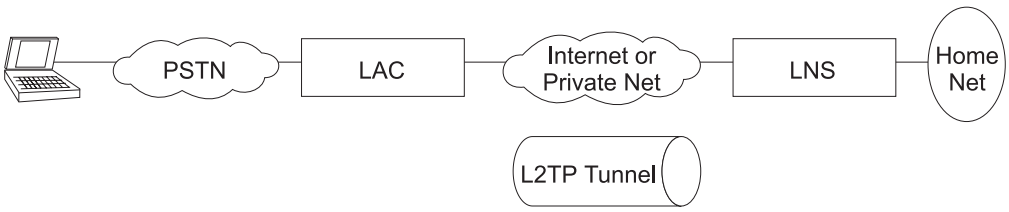



*Figure 49. Sample L2TP Network*

## L2TP Terms

The following terms are used when describing L2TP:

**Attribute Value Pair (AVP)**
A uniform method of encoding message types and bodies. This method maximizes the extensibility while permitting interoperability of L2TP.

**L2TP Access Concentrator (LAC)**
A device attached to one or more public service telephone network (PSTN) or ISDN lines capable of handling both PPP operation and the L2TP protocol. The LAC implements the media over which L2TP operates. L2TP

passes the traffic to one or more L2TP Network Servers (LNS). L2TP can tunnel any protocol carried by the PPP network.

**L2TP Network Server (LNS)**
An LNS operates on any platform that can be a PPP end station. The LNS handles the server side of the L2TP protocol. Because L2TP relies only on the single media over which L2TP tunnels arrive, the LNS can have only a single LAN or WAN interface, yet is still able to terminate calls arriving from any PPP interfaces supported by an LAC.

**Network Access Server (NAS)**
A device providing temporary, on-demand network access to users. This access is point-to-point using PSTN or ISDN lines.

**Session (Call)**
L2TP creates a session when an end-to-end PPP connection is attempted between a dial user and the LNS. The datagrams for the session are sent over the tunnel between the LAC and LNS. The LNS and LAC maintain the state information for each user attached to an LAC.

**Tunnel**
A tunnel is defined by an LNS-LAC pair. The tunnel carries PPP datagrams between the LAC and the LNS. A single tunnel can multiplex many sessions. A control connection operating over the same tunnel controls the establishment, release, and maintenance of all sessions and of the tunnel itself.

## Supported Features

L2TP runs over UDP/IP and supports the following functions:

- Tunneling of single user dial-in clients
- Tunneling of small routers, for example a router with a single static route to set up based on an authenticated user's profile
- Incoming calls to an LNS from an LAC
- Multiple calls per tunnel
- Proxy Authentication for PAP and CHAP
- Proxy LCP
- LCP restart in the event that Proxy LCP is not used at the LAC
- Tunnel end-point authentication
- Hidden AVP for transmitting a proxy PAP password
- Tunneling using a local rhelm (that is, user@rhelm) lookup table
- Tunneling using the PPP username lookup in the AAA subsystem

**Note:** Rhelm tunneling requires usernames in *name@rhelm* format. Tunneling this way requires the software to look through two tables to resolve the destination to which the dial-in user is tunnelled. The advantage of using this method of tunneling is that you need only define the rhelm and any usernames that match the rhelm will be tunnelled to the same destination.

User-based tunneling is resolved in a single table. It allows you the granularity of tunneling each user to a unique destination.

- BRS for an LNS (as a PPP end point)
- The ability to use the **delete interface** command to delete L2TP devices
- The ability to dynamically reconfigure L2TP devices

- Establishment of a sequencing, queueing, retransmission and flow control channel. L2TP also performs sequencing, queuing and flow control on data channels.

## Timing Considerations

The nature of tunneling PPP packets over routed networks creates some timing issues that you should consider. L2TP assumes that the connection between the LAC and LNS does not have a delay that is long enough to time out the tunneled peers. If the inter-peer latency repeatedly reaches or exceeds that of the PPP state machine's timeout (usually 3 seconds), then connectivity could be hindered. Note that if the latency between the LAC and LNS is this poor, then connectivity in general is so poor that the connection will be unreasonable even if the PPP state machines were kept alive artificially. If both sides possess the capability, then the PPP timeout may be extended to achieving connectivity over a very poor connection.

Besides latency, a bandwidth mismatch between the LAC/LNS pair and LAC/Client pair may cause problems. For instance, if the actual bandwidth between the LAC and LNS is significantly less than the bandwidth of the PPP client, then the LAC may spend significant time trying to send packets to the LNS. On the other hand, if the connection between the LNS and a host on the LNS home network is exceptionally fast compared with the dial-in client, then the LNS may be overburdened trying to send data to the LAC. L2TP implements a series of internal and external flow control techniques in an attempt to combat these situations.

## LCP Considerations

When using Proxy LCP, the LAC negotiates LCP and PPP continues processing at the LNS. The LAC forwards LCP options to the LNS so that the LNS is aware of what was negotiated. The LNS must remain flexible to the parameters negotiated by the client and LAC. If there are any parameters that are unacceptable to the LNS, then L2TP attempts to renegotiate LCP by sending an *LCP Configure Request* to the client across the tunnel.

The requirement for the LNS to remain flexible is of particular concern regarding the MRU. On the IBM LNS, the configured MRU is the maximum allowed for Proxy LCP. If the value in the Proxy LCP message from a LAC is greater than the MRU configured on the LNS, then L2TP will attempt to renegotiate LCP with an MRU equal to the configured MRU without changing other LCP options from the LAC.

## Configuring L2TP

To configure L2TP:

1. Access the L2TP feature using the **feature** command.

```
Config> feature layer-2-tunneling
Layer-2-Tunneling config>
```

2. Enable L2TP.

```
Layer-2-Tunneling config> enable l2tp
```

3. Add any L2TP networks needed. If this is to be strictly an LAC, you will not have to add any L2TP nets.

```
Layer-2-Tunneling Config>ADD L2-NETS
Additional L2 nets:  [0]? 10
Add unnumbered IP addresses for each L2 net? [Yes]: yes
```

```
Adding device as interface 31
Defaulting Data-link protocol to PPP
Adding device as interface 32
Defaulting Data-link protocol to PPP
Adding device as interface 33
Defaulting Data-link protocol to PPP
Adding device as interface 34
Defaulting Data-link protocol to PPP
Adding device as interface 35
Defaulting Data-link protocol to PPP
Adding device as interface 36
Defaulting Data-link protocol to PPP
Adding device as interface 37
Defaulting Data-link protocol to PPP
Adding device as interface 38
Defaulting Data-link protocol to PPP
Adding device as interface 39
Defaulting Data-link protocol to PPP
Adding device as interface 40
Defaulting Data-link protocol to PPP
```

4. Configure any L2TP tunnels needed.

   To configure a tunnel using an AAA local list:

```
Config>add tunnel-profile
Enter name:  []? lns.org
Enter hostname to use when connecting to this peer: []? lac.org
set shared secret? (Yes, No): [No] Y
Shared secret for tunnel authentication:
Enter again to verify:
Tunnel-Server endpoint address: [0.0.0.0]? 11.0.0.1

      PPP user name: lns.org
      Tunnel Server: 11.0.0.1
      Hostname: lac.org

 User 'lns.org' has been added
 Config>
```

   You can use the previous example to configure tunnel authorization on the LAC as well as “rhelm” tunneling in the form of “user@lns.org.”

   You can set tunnel authentication and authorization to be done at a particular RADIUS server. See “Using Authentication, Authorization, and Accounting (AAA) Security” on page 817.

   To tunnel by PPP username on a LAC using either an AAA local list or RADIUS:

```
Config>add ppp-user
Enter name:  []? peter
Password:
Enter again to verify:
Will 'peter' be tunneled? (Yes, No): [No] Y
Enter hostname to use when connecting to this peer: []? lac.org
Tunnel-Server endpoint address: [0.0.0.0]? 11.0.0.1

     PPP user name: peter
     Tunnel Server: 11.0.0.1
     Hostname: lac.org

Is information correct? (Yes, No, Quit): [Yes]

User 'peter' has been added
Config>
```

5. Configure the various L2TP parameters using the **set** command, if desired.
6. Configure the PPP parameters for all of the L2 nets using the encapsulator command, if desired.

```
Layer-2-Tunneling Config>encapsulator
PPP-L2TP Config>
```

   When you have completed the PPP configuration, enter **exit** to return to the L2TP configuration environment.

7. Enable any L2TP functions using the **enable** command.

# Chapter 56. Configuring and Monitoring L2TP

This chapter describes the L2TP Protocol configuration and operational commands. Sections in this chapter include:



## L2TP Configuration Commands

Table 96 summarizes the L2TP configuration commands and the rest of this section explains the commands. Enter these commands at the `L2TP Config>` prompt.

*Table 96. L2TP Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Add | Adds L2TP nets or peers. |
| Delete | Deletes L2TP peers from the configuration. |
| Disable | Disables L2TP and L2TP functions. |
| Enable | Enables L2TP or L2TP functions. |
| Encapsulator | Allows you to configure PPP parameters for all of the L2TP nets. |
| List | Displays information about the various L2TP configuration. |
| Set | Allows you to set buffers, the call receive window, and other L2TP parameters. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Add

Use the **add** command to add an L2TP peer (LAC or LNS) or an L2-Net. One L2-Net is required for each concurrent PPP session that ends on this router. The end of a tunneled PPP session is the LNS end point of the tunnel.

**Syntax: add**
L2-nets

"Configuring L2TP" on page 663 contains an example of the **add** command.

**L2-nets**

**Note:** This command can be entered entirely in lower case. The initial character is shown in upper case for clarity.

Adds an L2-Net to the L2TP configuration. One L2-Net is required for each concurrent PPP session that is to be terminated at this router. If this router is to be used strictly as an LAC, no virtual L2-Nets are necessary. When you enter this command, you are prompted for the number of additional nets and whether to add unnumbered IP addresses for each L2 net.

The number of additional nets refers to how many nets L2TP automatically adds at this time. These nets are in addition to any L2-Nets that may already exist.

Adding unnumbered IP addresses for each L2-Net automatically add unnumbered IP entries into the IP routing table for each of the L2-Nets. Unnumbered IP addresses are the preferred mode of operation. If you need numbered addresses for the L2-Nets, you can alter them in the IP protocol configuration environment (refer to the chapter entitled “Configuring IP” in the *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1*).

## Disable

Use the **disable** command to disable L2TP functions or disable L2TP itself.

**Syntax:** disable
- call-rcv-window
- force-chap-challenge
- hiding-for-pap-attributes
- L2tp
- proxy-auth
- proxy-lcp
- tunnel-authentication

**call-rcv-window**
L2TP can queue packets for each call in order to perform sequencing and congestion control. Each call has its own queue, or window, whose size must be transmitted to the peer for the flow control algorithms to work correctly. Disabling the *call-rcv-window* turns off all flow control for each session. This may be desirable when the connection between the LAC and LNS is known to be of high quality, sufficient bandwidth, and not prone to a great deal of packet reordering.

**force-chap-challenge**
Disables the LNS CHAP rechallenge of a client. You may need to disable the CHAP rechallenge if the PPP client has difficulty with CHAP rechallenges.

**hiding-for-pap-attributes**
Disables the encryption of Proxy PAP information between the LAC and LNS.

**L2tp**

**Note:** This command can be entered entirely in lower case. The initial character is shown in upper case for clarity.

Disables L2TP on this router.

**proxy-auth**
Disables sending PPP proxy-authentication from LAC to LNS.

**proxy-lcp**
Disables sending LCP information from LAC to LNS.

**tunnel-authentication**
Disables peer authentication based on a shared secret for all tunnels.

## Enable

Use the **enable** command to enable L2TP functions or enable L2TP itself.

**Syntax:**

enable

- force-chap-challenge
- hiding-for-pap-attributes
- L2tp
- proxy-auth
- proxy-lcp
- tunnel-authentication

**force-chap-challenge**
Enables the LNS CHAP rechallenge of a client even if the LNS receives a proxy CHAP. This is preferable from a security standpoint, if it is known that the client can handle such a rechallenge without problems.

**hiding-for-pap-attributes**
Enables the encryption of Proxy PAP information between the LAC and LNS.

**L2tp**

**Note:** This command can be entered entirely in lower case. The initial character is shown in upper case for clarity.

Enables L2TP on this router.

**proxy-auth**
Enables sending PPP proxy-authentication from LAC to LNS.

**proxy-lcp**
Enables sending LCP information from LAC to LNS.

**tunnel authentication**
Enables peer authentication based on a shared secret for all tunnels.

## Encapsulator

Use the **encapsulator** command to configure the PPP parameters for the L2-Nets.

**Syntax:** encapsulator

## List

Use the **list** command to display the state of the various L2TP configuration parameters.

**Syntax:** list

```
Layer-2-Tunneling Config>list
GENERAL ADMINISTRATION
------- --------------
  L2TP                                = Enabled
  Maximum number of tunnels           = 20
  Maximum number of calls (total)     = 50
  Buffers Requested                   = 300

CONTROL CHANNEL SETTINGS
------- ------- --------
  Tunnel Auth                         = Enabled
```

```
  Tunnel Rcv Window                    = 4
  Retransmit Retries                   = 6
DATA CHANNEL SETTINGS
---- ------- --------
  Force CHAP Challenge (extra security)= Disabled
  Hiding for PAP Attributes            = Disabled
  Call Rcv Window                      = 6

MISCELLANEOUS
-------------
  SEND PROXY-LCP FROM LAC              = Enabled
  SEND PROXY-AUTH FROM LAC             = Enabled
```

## Set

Use the set command to configure the L2TP operational parameters.

**Syntax:** set buffers

call-rcv-window

max-calls

max-tunnels

transmit-retries

tunnel-rcv-window

**buffers**
Specifies the number of requested internal L2TP buffers. If there is not enough memory to satisfy the request, only a portion of the buffers will be available upon reboot. To confirm the amount of memory while L2TP is active, use the **memory** command (see “Memory” on page 674).

**Valid values:** 1 to 1000

**Default value:** 200

**call-rcv-window**
Specifies the number of packets to be used as a receive window and enables the call-rcv-window. If flow control is enabled on the data channel, a receive window size must be designated, both for use by the protocol on this router and for communication to the peer using start-up messages. The value configured is for all calls initiated by this router.

**Valid values:** 0 to 100

**Default value:** 6

**max-calls**
Specifies the maximum number of calls across all tunnels that can be active at a given time either as LAC or LNS.

**Valid values:** 1 to 500

**Default value:** 100

**max-tunnels**
Specifies the maximum number of tunnels that can be active at a given time either as LAC or LNS.

**Valid values:** 1 to 100

**Default value:** 30

**transmit-retries**
Specifies the number of times a packet is retransmitted on the control channel before the session or tunnel is declared inactive and is shut down.

**Valid values:** 2 to 100

**Default value:** 6

**tunnel-rcv-window**
Specifies the receive window size for the reliable control connections transport. This transport transmits and receives the messages necessary for tunnel or session setup, tear down, and maintenance.

**Valid values:** 1 to 100

**Default value:** 4

## Accessing the L2TP Monitoring Prompt

To access the L2TP monitoring prompt:

1. Enter **talk 5** at the OPCON (*) prompt.
2. Enter **feature layer-2-tunneling** at the GWCON (+) prompt.

## L2TP Monitoring Commands

This section summarizes and then describes the L2TP monitoring commands. Enter the commands at the `Layer-2-Tunneling Console>` prompt.

Table 97 summarizes the L2TP monitoring commands.

*Table 97. L2TP Monitoring Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Call | Displays statistics and information about each call in progress. |
| Kill | Ends a call or tunnel immediately. |
| Memory | Displays the current L2TP buffer allocation and use. |
| Start | Starts a tunnel with another peer. |
| Stop | Stops a call or tunnel and allows each peer to perform any needed administration. |
| Tunnel | Displays statistics and information on each existing tunnel. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Call

Use the **call** command to display call statistics and information.

**Syntax:** call errors
physical-errors
queue
state
statistics

**errors** Displays the general transmission errors that occurred on the calls.

**Example:**

```
Layer-2-Tunneling Console> call errors
CallID | Serial # | ACK-timeout | Dropped pkts
 56744 |        1 |           0 |            0
```

**CallID** The local identifier associated with this call.

**Serial #**
The number used for logging this call.

**ACK-timeout**
The number of times a timeout notification has been received from the peer.

**Dropped pkts**
The number of packets that have been declared lost for this call. These are packets which should have been received, but were signalled as lost by the peer.

**physical-errors**
Displays the data errors that occurred on the calls.

**Example:**

```
Layer-2-Tunneling Console> call physical-errors
CallID | Serial# | CRC    |framing|  HW    | buffer|timeout| align-| time since
       |         | Errors| Errors|overrun|overrun| Errors| ment  | updated
 56744 |       1 |     0 |     0 |     0 |     0 |     0 |     0 |
```

**CallID** The local identifier associated with this call.

**Serial #**
The number used for logging this call.

**CRC Errors**
The number of packets on which the CRC did not match.

**framing errors**
The number of packets with a framing error.

**HW overrun**
The number of times a hardware overrun occurred.

**buffer overrun**
The number of times a buffer overrun occurred.

**timeout errors**
The number of times an interface timed out.

**alignment**
The number of times an alignment error occurred.

**time since updated**
The elapsed time since last poll for errors.

**queue** Displays information about the queue for each call.

**Example:**

```
Layer-2-Tunneling Console> call queue
CallID | Serial # |Tx Win|Rx Win|   Ns   |   Nr   |Rx Q|Tx Q|priority| out Q
 56744 |        1 |    4 |    4 |    100 |   200  |  0 |  0 |      0 |    0
```

**CallID** The local identifier associated with this call.

**Serial #**
The number used for logging this call.

**Tx Win**
The peer's maximum receive window for data.

**Rx Win**
The local maximum transmit window.

**Ns** The next packet sequence number to send for this call.

**Nr** The next packet sequence number expected to be received for this call.

**Rx Q** The current number of packets on the receive queue.

**Tx Q** The current number of packets on the transmit queue.

**priority**
The number of priority PPP packets waiting to be transmitted by L2TP.

**out Q** The number of regular PPP packets waiting to be transmitted by L2TP.

**state** Displays the current state of each call.

**Example:**

```
Layer-2-Tunneling Console> call state
CallID | Serial # | Net #  |    State    | Time Since Chg | PeerID | TunnelID
 56744 |        1 |     2  | Established |   00:00:00     |  345   | 45678
```

**CallID** The local identifier associated with this call.

**Serial #**
The number used for logging this call.

**Net #** The device number associated with this call. For an LNS call, this is the L2-Net. For an LAC call, this is the PPP device that received the initial call.

**State** The current call state. Valid call states are:

**Established**
Ready for tunneled network traffic.

**Idle** The call is idle.

**Wait Cs Answer**
Waiting for the communication link to open.

**Wait Reply**
Waiting for a reply from the peer.

**Wait Tunnel**
Waiting for tunnel establishment.

**Time since chg**
The elapsed time since the last state change.

**PeerID**
The Peer's call ID.

**TunnelID**
The local tunnel associated with this call.

**statistics**
Displays statistics about the data transmission for each call.

**Example:**

```
Layer-2-Tunneling Console> call statistics
CallID | Serial # | Tx Pkts | Tx Bytes | Rx Pkts | Rx Bytes |  RTT  | ATO
 56744 |        1 |     34  |    1056  |     45  |    1567  |   10  | 34
```

**CallID** The local identifier associated with this call.

**Serial #**
The number used for logging this call.

**Tx Pkts**
The number of packets transmitted for this call.

**Tx Bytes**
The number of bytes transmitted for this call.

**Rx Pkts**
The number of packets received for this call.

**Rx Bytes**
The number of bytes received for this call.

**RTT** The currently calculated round trip time for this call.

**ATO** The currently calculated adaptive time out for this call.

## Kill

Use the **kill** to immediately end a tunnel. This command releases all of the local resources for a tunnel thereby forcing the end of the connection. No notification of the end of the tunnel is sent to the peer.

**Note:** Use this command only if the **stop** command is unable to end a tunnel.

**Syntax:** kill tunnel *tunnelid*

**tunnel** *tunnelid*
Specifies the tunnel to end.

## Memory

Use the **memory** command to display L2TP's current memory utilization.

**Syntax:** memory

**Example:**

```
Layer-2-Tunneling Console> mem
Number of layer-2-tunneling buffers: Requested = 2000, Total = 1200, Free
= 1000
```

In this example, you configured 2000 buffers but were able to allocate only 1200. Currently, 200 buffers are in use leaving 1000 free.

## Start

Use the **start** command to start a tunnel with another peer.

**Syntax:** start (no parameters will prompt for hostname)

**tunnel** *hostname*

**hostname**
The name of the host with which L2TP establishes the tunnel.

## Stop

Use the **stop** command to stop a tunnel. Any required cleanup is completed before the tunnel ends.

**Syntax:** stop tunnel *tunnelid*

**tunnel** *tunnelid*
Specifies the tunnel to end.

## Tunnel

Use the **tunnel** command to display statistics and information about all tunnels.

**Syntax:** tunnel
call
errors
peer
queue
state
statistics
transport

**calls** Displays all tunnels and the call state for each call within each tunnel.

**errors** Displays the errors that have occurred on a tunnel.

**Example:**

```
Layer-2-Tunneling Console> tunnel errors
Tunnel ID │ Type │ACK-timeouts
 96785    │ L2TP │    0
```

**Tunnel ID**
The local identifier associated with a tunnel.

**Retransmissions**
The number of packets that were retransmitted on the tunnel.

**peer** Displays the tunnels and the peers associated with the tunnels.

**Example:**

```
Layer-2-Tunneling Console> tunnel peer
Tunnel ID │ Type │ Peer ID │ Peer Hostname
 96785    │ L2TP │   89777 │ mypeer
```

**Tunnel ID**
The local identifier associated with a tunnel.

**Peer ID**
The peer's tunnel identifier assigned to this tunnel.

**Peer Hostname**
The hostname of the peer as it appears in the local database.

**queue** Displays information about the queue for each tunnel.

**Example:**

```
Layer-2-Tunneling Console> tunnel queue
Tunnel ID │ Type │ Rx Win │ Tx Win │   Ns   │   Nr   │ Rx Q │ Tx Q
 96785    │ L2TP │      4 │      4 │      5 │      6 │    0 │    0
```

**Tunnel ID**
The local identifier associated with a tunnel.

**Rx Win**
The local maximum number of packets that constitute the receive window.

**Tx Win**
The peer's maximum number of packets that constitute the receive window.

**Ns** The sequence number of the next packet to send.

**Nr** The sequence number of the next packet to receive.

**Rx Q** The number of packets currently on the receive queue.

**Tx Q** The number of packets currently on the transmit queue.

**state** Displays the current state of all the tunnels.

**Example:**

```
Layer-2-Tunneling Console> tunnel state
Tunnel ID │ Type │ Peer ID │    State    │ Time Since Chg │ # Calls │ Flags
 96785    │ L2TP │  89777  │ Established │   00:00:00     │       1 │  0
```

**Tunnel ID**
The local identifier associated with a tunnel.

**Peer ID**
The peer's tunnel identifier assigned to this tunnel.

**State** The current tunnel state. Valid tunnel states are:

**Established**
The tunnel is established.

**Idle** The tunnel is idle.

**Wait Ctrl Reply**
The host is waiting for a reply from the peer.

**Wait Ctrl Conn**
The host is waiting for a connection indication.

**Time since chg**
The elapsed time since the last state change.

**# Calls**
The number of active calls on this tunnel.

**Flags** The flags used to control the connection messages on this tunnel.

**statistics**
Displays the statistics associated with the tunnels.

**Example:**

```
Layer-2-Tunneling Console> tunnel statistics
Tunnel ID │ Type │ Tx Pkts │ Tx Bytes │ Rx Pkts │ Rx Bytes │  RTT  │ ATO
 96785    │ L2TP │       4 │       78 │       5 │       89 │    10 │ 31
```

**Tunnel ID**
The local identifier associated with a tunnel.

**Tx Pkts**
The number of packets transmitted.

**Tx Bytes**
The number of bytes transmitted.

**Rx Pkts**
The number of packets received.

**Rx Bytes**
The number of bytes received.

**RTT** The currently calculated round trip time for tunnel control connection messages.

**ATO** The currently calculated adaptive timeout for tunnel control connection messages.

**transport**

Displays UDP information about the tunnels.

**Example:**

```
Layer-2-Tunneling Console> tunnel transport
Tunnel ID │ Type │ Peer IP Address │ UDP Src │ UDP Dest
 96785    │ L2TP │  11.0.0.102     │  1056   │   1089
```

**Tunnel ID**

The local identifier associated with a tunnel.

**Peer IP address**

The peer's IP address for this tunnel.

**UDP Src**

The UDP source port for this tunnel.

**UDP Dest**

The UDP destination port for this tunnel.

# Part 4. Understanding, Configuring and Using Features

# Chapter 57. Using Bandwidth Reservation and Priority Queuing

This chapter describes the Bandwidth Reservation System and priority queuing features currently available for Frame Relay and PPP interfaces. It includes the following sections:



## Bandwidth Reservation System

The Bandwidth Reservation System (BRS) allows you to decide which packets to drop when demand (traffic) exceeds supply (throughput) on a network connection. When bandwidth utilization reaches 100%, BRS determines which traffic to drop based on your configuration.

Bandwidth reservation ″reserves″ transmission bandwidth for specified classes of traffic. Each class has an allocated minimum percentage of the connection's bandwidth. See Figure 50 on page 682 and Figure 51 on page 682.

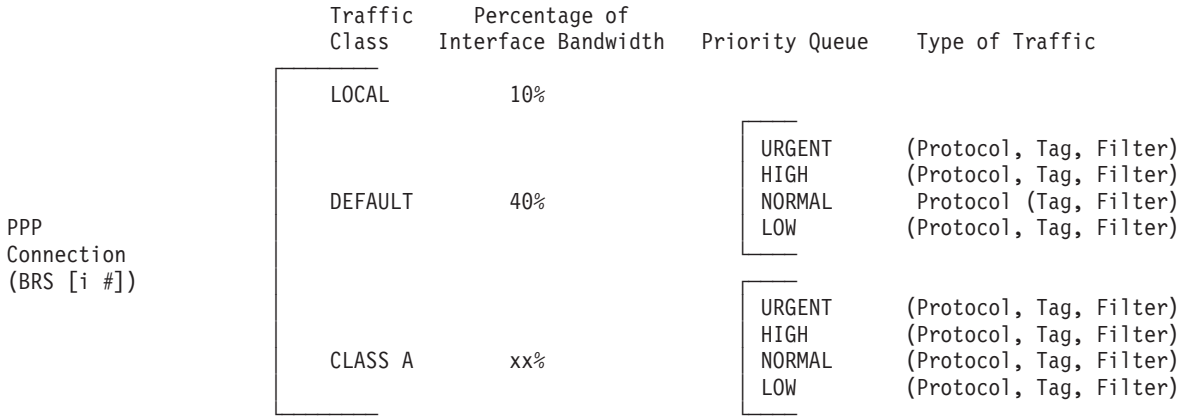
On PPP interfaces, you define traffic classes (t-classes) and each traffic class is allocated a percentage of the PPP interface's bandwidth. There are at least two traffic classes:

1. A LOCAL class which is allocated bandwidth for packets that are locally originated by the router (e.g. IP RIP packets)
2. A DEFAULT class to which all other traffic is initially assigned.

You can create additional traffic classes and assign protocols, filters and tags to the priority queues within a traffic class. See Figure 50 on page 682.

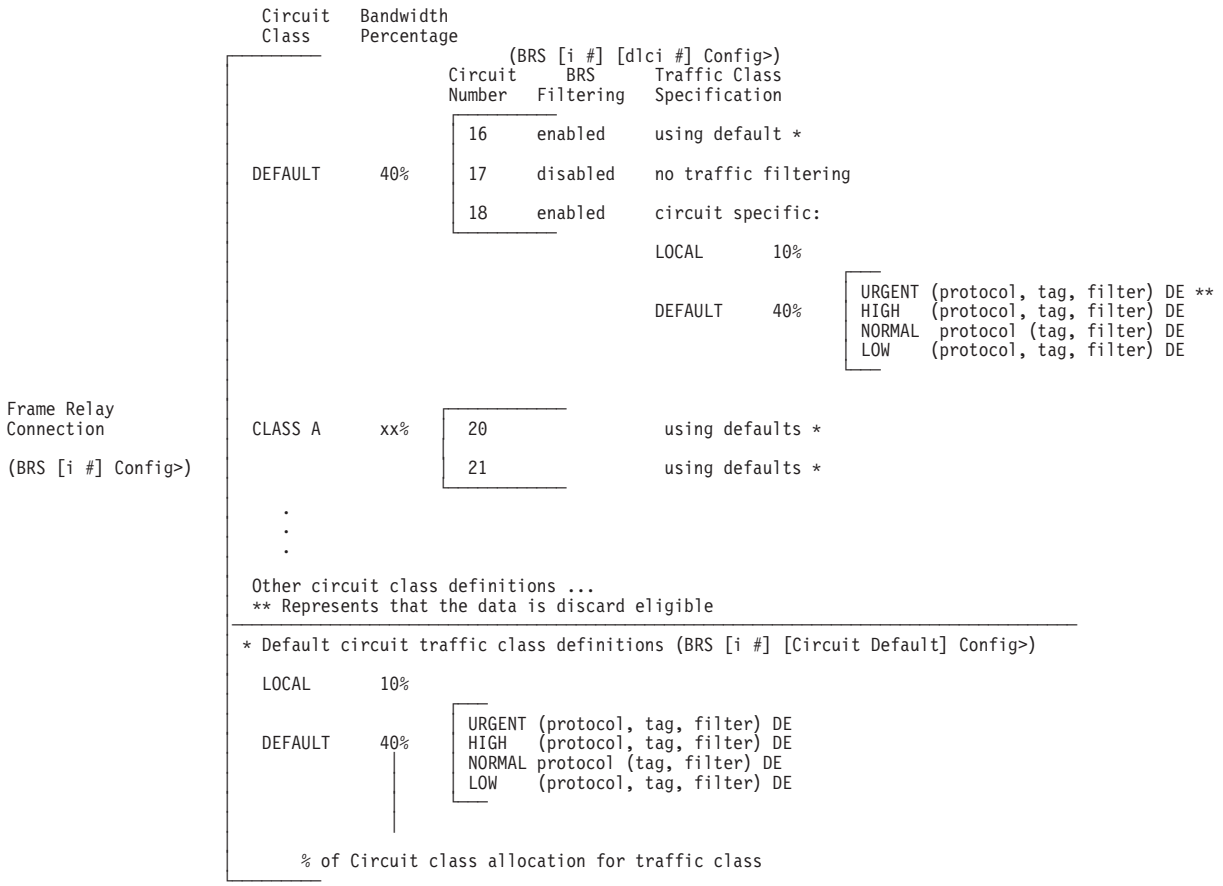
On Frame Relay interfaces, you define circuit classes (c-classes) and each circuit class is allocated a percentage of the Frame Relay interface's bandwidth. There is at least one circuit class: the DEFAULT circuit class to which all circuits are initially assigned. You can create additional circuit classes and assign circuits to these c-classes. On each Frame Relay circuit, you can define traffic classes (t-classes) and each traffic class is allocated a percentage of the Frame Relay circuit's bandwidth. The traffic class support for Frame Relay circuits is analogous to the traffic class support for PPP interfaces. See Figure 51 on page 682 for the Frame Relay Circuit Class and Traffic Class Relationships.

## Using BRS and Priority Queuing




**Note:** All protocols are initially assigned to the NORMAL priority queue of the DEFAULT traffic class. You can assign a protocol, filter, or tag to any priority queue within a traffic class.

*Figure 50. PPP BRS Traffic Class and Traffic Class Priority Queue Relationship*




**Note:** All protocols are initially assigned to the NORMAL priority queue of the DEFAULT traffic class. You can assign a protocol, filter, or tag to any priority queue within a traffic class.

*Figure 51. Frame Relay BRS Circuit Class and Traffic Class Relationship*

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 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 device 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 connection types:

- Frame Relay (serial line or dial circuit interface)
- PPP (serial line or dial circuit interface)

## 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 58. Configuring and Monitoring Bandwidth Reservation” on page 699 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 do not 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.

## Discard Eligibility

The Frame Relay network may discard transmitted data exceeding CIR on a PVC. The DE bit can be set by the router to indicate that some traffic should be considered discard eligible. If appropriate, the Frame Relay network will discard frames marked as discard eligible, which may allow frames that are not marked discard eligible to make it through the network. When assigning a protocol, filter, or tag to a traffic class, you can specify whether or not the protocol, filter, or tag traffic is discard eligible. See 705 for more information on how to configure traffic as discard eligible.

## Default Circuit Definitions for Traffic Class Handling

Frame Relay interfaces can have many circuits defined. Rather than having to fully configure traffic class definitions for each circuit, BRS allows you to define a default set of traffic classes and protocol, filter, and tag assignments called default circuit definitions that can be used by any circuit on the interface. When BRS is initially enabled on a circuit, the circuit is initialized to use default circuit definitions. If a circuit cannot use the default circuit definitions for traffic class handling then you can create circuit specific definitions by using the **add-class**, **change-class**, **assign**, **deassign**, **tag**, and **untag** commands.

If a circuit is using circuit specific definitions and you want it to use the default circuit definitions instead, you can use the **use-circuit-defaults** command at the circuit's BRS prompt.

The default circuit definitions for traffic class handling are defined by using the **set-circuit-defaults** at the BRS Frame Relay interface prompt. This command gets you to a BRS circuit defaults prompt where you can add, change, and delete traffic classes, assign and deassign protocols, filters, and tags, and create BRS tags. Changes to the default circuit definitions for traffic classes result in dynamic updates to the traffic class handling for all circuits using the default circuit definitions.

## 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 connections. See “Queue-length” on page 715 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.

## Priority Queuing Without Bandwidth Reservation

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.

## Configuring 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
- HPR/IP

### BRS Filters

Using bandwidth reservation, you can treat specific protocol traffic differently from other traffic that is using the same protocol type. For example, you can assign SNMP IP traffic to a different traffic class and priority than other IP traffic. In this example, SNMP is a BRS filter because it ″filters″ (i.e. uniquely identifies) specific protocol traffic. IP, ASRT (bridging) and APPN-HPR protocol traffic can be ″filtered″ by bandwidth reservation and the following filters are supported:

- IP tunneling
- SDLC tunneling over IP (SDLC Relay)
- Rlogin
- Telnet
- SNA/APPN-ISR
- APPN-HPR
- SNMP
- IP Multicast
- DLSw
- MAC Filter
- NetBIOS
- Network-HPR
- High-HPR
- Medium-HPR
- Low-HPR
- XTP
- TCP/UDP port numbers or sockets

## BRS and Filtering

The following sections describe how to use BRS with various types of filtering.

## 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. See “Chapter 59. Using MAC Filtering” on page 723 for additional information about MAC filtering.

**Note:** Tags can be applied ***only*** to bridged packets. On a PPP or Frame Relay connection, 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 59. Using MAC Filtering” on page 723 and “Chapter 60. Configuring and Monitoring MAC Filtering” on page 727.

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. Refer to “Chapter 59. Using MAC Filtering” on page 723 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.

## TCP/UDP Port Number Filtering

You can assign TCP/IP packets from a range of TCP or UDP ports to a BRS t-class and priority based on the packet’s UDP or TCP port number and, optionally, upon a socket. You can specify up to 5 UDP/TCP port number filters, where the filters specify either an individual TCP or UDP port number, a range of TCP or UDP port numbers, or a socket identifier (combination of port number and IP address). You can then assign that filter to a BRS traffic class and priority within the class.

If UDP/TCP port filtering is enabled, BRS looks at each TCP or UDP packet and checks to see if the destination or source port number matches one of the port numbers you have specified for filtering. Also, if you define an IP address as part of the BRS UDP/TCP filter and the destination or source IP address matches the filter address you define, BRS assigns the packet to the traffic class and priority for that port number filter.

For example, you can configure a UDP port number filter for UDP port numbers in the range 25 to 29 and assign the filter to traffic class ’A’ with a priority of ’normal’. BRS queues any UDP packets with a source or destination port number from 25 to 29 on the normal priority queue for traffic class ’A’.

You can also configure a TCP port number filter for TCP port number 50 for IP address 5.5.5.25 and assign the filter to traffic class ’B’ with priority ’urgent’. BRS queues any TCP packets whose source or destination port number is 50 and whose destination or source IP address is 5.5.5.25 on the urgent priority queue for traffic class ’B’.

## Using IP Version 4 Precedence Bit Processing for SNA Traffic in IP Secure Tunnels and Secondary Fragments

BRS normally differentiates IP TCP and UDP traffic according to its port numbers. However, BRS cannot identify the ports of traffic that have been encapsulated twice, such as IP traffic transported through an IP secure tunnel or are in a secondary UDP or TCP fragments. As a result, BRS cannot filter these kinds of traffic. IP

version 4 precedence bit processing allows BRS to continue to filter encapsulated SNA traffic that is transported through an IP secure tunnel or are in a secondary TCP or UDP fragment.

When APPN/HPR traffic is being routed over IP, each transmission priority of APPN-HPR (network, high, medium, and low) is mapped to a particular value of the three IP version 4 precedence bits.

- The HPR network transmission priority maps to the IPv4 precedence value of '110'b.
- The HPR high transmission priority maps to the IPv4 precedence value of '100'b.
- The HPR medium transmission priority maps to the IPv4 precedence value of '010'b.
- The HPR low transmission priority maps to the IPv4 precedence value of '001'b.

When IPv4 precedence filtering is enabled for BRS and the precedence bits in an IP packet match one of the values used for APPN/HPR traffic, then the packet is queued on the priority queue of the BRS t-class to which the corresponding HPR transmission priority is assigned. For example, if an IP packet has a precedence value of '110'b and the BRS HPR-Network filter is assigned to t-class A and priority level normal, then the packet is queued on the normal priority queue of t-class A. If a BRS HPR transmission priority filter is not configured, but the APPN-HPR filter is configured, then the packet is queued on the priority queue and t-class to which the APPN-HPR filter is assigned.

These three kinds of traffic map to the IPv4 precedence value '011'b:

- APPN/HPR XID traffic that is sent when APPN/HPR is routed over IP
- DLSw traffic
- TN3270 traffic

Because several types of traffic map to one value, BRS cannot distinguish between them when it is enabled to filter based on the IPv4 precedence bits. Therefore, when BRS encounters an IP packet with a precedence value of '011'b, it evaluates the BRS filters in the following order to determine whether or not the filter is enabled. When it finds a BRS filter that is configured, the packet is queued on the priority queue and t-class to which the BRS filter is assigned:

- SNA/APPN-ISR (used for APPN/HPR XID exchanges)
- DLSw
- Telnet

If a packet has one of the precedence values that are filtered by BRS, but none of the applicable BRS filter types are configured, the packet is queued on the priority queue and the BRS t-class to which the IP protocol is assigned.

When TN3270 traffic is sent by a client to the 2216 over a wide-area network where BRS is enabled, traffic from the client cannot be prioritized by BRS unless the client sets the precedence bits to '011'b.

You must configure IPv4 precedence bit handling in multiple places:

1. In BRS you configure whether or not BRS should filter based on the IPv4 precedence bits. It only performs this type of filtering for IP secure tunnel packets or TCP and UDP secondary fragment packets.
2. When you configure DLSw, HPR over IP, and TN3270, you specify whether or not the 2216 should set the IPv4 precedence bits for packets that it originates for each of these protocol types.

Perform these three steps to use IPv4 precedence bit filtering:

1. Activate IPv4 precedence filtering in BRS.
2. Configure BRS t-classes and assign protocols and filters for various categories of SNA traffic, as you would for SNA traffic that is not transported in an IP secure tunnel or is not fragmented.
3. Enable the setting of the IPv4 precedence bits when configuring the DLSw, HPR over IP, and TN3270 protocols.

## SNA and APPN Filtering for Bridged Traffic

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 bridged 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 in 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 X'04', X'08', X'0C', or X'C8' 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.

APPN-HPR traffic that is being routed over IP is filtered using the UDP port number assigned for network, high, medium and low HPR transmission priorities. An additional UDP port number is used for XID exchanges. All of the UDP port numbers used to support APPN-HPR over IP are configurable.

If APPN is not enabled in an intermediate router in the IP network, you can configure UDP port numbers for HPR over IP from the `BRS Config>` command prompt. If APPN is enabled in the device, BRS will use the values configured at the `APPN Config>` command prompt.

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 Filtering Precedence

It is possible for a packet to match more than one BRS filter type. For example, an IP tunneled bridge packet containing SNA data would match the IP tunneling filter and the SNA/APPN-ISR filter. The order in which the filters are evaluated to determine whether or not a packet matches a BRS filter type is as follows:

1. MAC filter tag match for bridging packets (IP/ASRT)
2. NetBIOS for bridging (IP/ASRT)
3. SNA/APPN-ISR for bridging (IP/ASRT)
4. HPR-Network (IP/ASRT/APPN-HPR)
5. HPR-High (IP/ASRT/APPN-HPR)
6. HPR-Medium (IP/ASRT/APPN-HPR)
7. HPR-Low (IP/ASRT/APPN-HPR)
8. APPN-HPR (IP/ASRT)
9. UDP/TCP port number filters (IP)
10. IP tunneling (IP)
11. SDLC relay (IP)
12. DLSw (IP)
13. Multicast (IP)
14. SNMP (IP)
15. Rlogin (IP)
16. Telnet (IP)
17. XTP (IP)

**Note:** The protocols for which a filter applies are shown in parentheses

# Sample Configurations

## Using Default Circuit Definitions for Traffic Class Handling of Frame Relay Circuits

**Notes:**

1. Configure feature BRS.
2. Enable BRS on interface 1.
3. Enable BRS on circuits 16, 17, 18. Default circuit definitions for traffic class handling are used for these circuits.
4. Access the set-circuit-defaults menu to define default circuit definitions for traffic class handling.
5. Add traffic classes and assign protocols and filters to the traffic classes.
6. List and show the BRS definitions for circuit 16. Since circuit 16 is using default circuit definitions, the traffic classes and protocol and filter assignments defined by the default circuit definitions are displayed.
7. Change circuit 17 from using default circuit definitions to use circuit-specific definitions for traffic class handling by creating a unique class, CIRC171. This class can have protocols, filters, or tags assigned to it.

8 Change the default circuit definitions such that the DEF1 and DEF2 traffic classes each reserve 10% of the bandwidth and then show that these changes are picked up by circuit 16 but not by circuit 17, since circuit 17 is now using circuit-specific definitions.

9 Alter circuit 17 to use default circuit definitions for traffic class handling instead of circuit-specific definitions.

```
t 6
 Gateway user configuration
 Config>feature brs     1
 Bandwidth Reservation User Configuration
 BRS Config>interface 1    2
 BRS [i 1]Config>enable
 Please reload router for this command to take effect.
 BRS [i 1] Config>circuit 16   3
 BRS [i 1][dlci 16] Config>enable
 Defaults are in effect for this circuit.
 Please reload router for this command to take effect.
 BRS [i 1][dlci 16] Config>exit
 BRS [i 1]Config>circuit 17
 BRS [i 1][dlci 17] Config>enable
 Defaults are in effect for this circuit.
 Please reload router for this command to take effect.
 BRS [i 1][dlci 17] Config>exit
 BRS [i 1]Config>circuit 18
 BRS [i 1][dlci 18] Config>enable
 Defaults are in effect for this circuit.
 Please reload router for this command to take effect.
 BRS [i 1][dlci 18] Config>
 *reload
 Are you sure you want to reload the gateway? (Yes or [No]): yes


 *t 6
 Gateway user configuration
 Config>feature brs
 Bandwidth Reservation User Configuration
 BRS Config>interface 1
 BRS[i 1] Config>list

 BANDWIDTH RESERVATION listing from SRAM
 bandwidth reservation is enabled
 interface number 1
 maximum queue length 10, minimum queue length 3
 total bandwidth allocated 10%
 total circuit classes defined (counting one default) 1


 class DEFAULT has 10% bandwidth allocated
   the following circuits are assigned:
       16  using defaults.
       17  using defaults.
       18  using defaults.

 default class is DEFAULT


BRS [i 1] Config>?
 ENABLE
 DISABLE
 SET-CIRCUIT-DEFAULTS
 CIRCUIT
 ADD-CIRCUIT-CLASS
 DEL-CIRCUIT-CLASS
 CHANGE-CIRCUIT-CLASS
 DEFAULT-CIRCUIT-CLASS
 ASSIGN-CIRCUIT
 DEASSIGN-CIRCUIT
 QUEUE-LENGTH
 LIST
 SHOW
 CLEAR-BLOCK
 EXIT
 BRS [i 1] Config>set-circuit-defaults    4
 BRS [i 1] [circuit defaults] Config>?
 ADD-CLASS
```

```
DEL-CLASS
CHANGE-CLASS
DEFAULT-CLASS
TAG
UNTAG
ASSIGN
DEASSIGN
LIST
EXIT
BRS [i 1] [circuit defaults] Config>add   5
Class name [DEFAULT]?DEF1
Percent bandwidth to reserve [10]? 5
BRS [i 1] [circuit defaults] Config>add
Class name [DEFAULT]?DEF2
Percent bandwidth to reserve [10]?5
BRS [i 1] [circuit defaults] Config>assign ip
Class name [DEFAULT]?DEF1
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]?
Frame Relay Discard Eligible <NO/YES> [NO]?
BRS [i 1] [circuit defaults] Config>assign asrt
Class name [DEFAULT]? DEF2
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]?
Frame Relay Discard Eligible <NO/YES> [NO]?
BRS[i 1] [circuit defaults] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, default circuit
total bandwidth allocated 60%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ARP with default priority is not discard eligible
    protocol DNA with default priority is not discard eligible
    protocol VINES with default priority is not discard eligible
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible

class DEF1 has 5% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL is not discard eligible


class DEF2 has 5% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ASRT with priority NORMAL is not discard eligible

assigned tags:

default class is DEFAULT with priority NORMAL

BRS  [i 1] [circuit defaults] Config>exit
BRS  [i 1] Config>circuit 16    6
BRS  [i 1][dlci 161] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, circuit number 16 using defaults.
maximum queue length 10, minimum queue length 3
total bandwidth allocated 60%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ARP with default priority is not discard eligible
    protocol DNA with default priority is not discard eligible
    protocol VINES with default priority is not discard eligible
```

```
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible

 class DEF1 has 5% bandwidth allocated
   the following protocols and filters are assigned:
     protocol IP with priority NORMAL is not discard eligible


 class DEF2 has 5% bandwidth allocated
   the following protocols and filters are assigned:
     protocol ASRT with priority NORMAL is not discard eligible

 assigned tags:

 default class is DEFAULT with priority NORMAL


BRS  [i 1] [dlci 16] Config>show

 BANDWIDTH RESERVATION currently in RAM
 interface number 1, circuit number 16 using defaults.
 maximum queue length 10, minimum queue length 3
 4 current defined classes:
   class LOCAL has 10% bandwidth allocated
   class DEFAULT has 40% bandwidth allocated
   class DEF1 has 5% bandwidth allocated
   class DEF2 has 5% bandwidth allocated

 protocol and filter assignments:

 Protocol/Filter      Class              Priority    Discard Eligible
 ---------------      -----              --------    ----------------
 IP                   DEF1                 NORMAL             NO
 ARP                  DEFAULT              NORMAL             NO
 DNA                  DEFAULT              NORMAL             NO
 VINES                DEFAULT              NORMAL             NO
 IPX                  DEFAULT              NORMAL             NO
 OSI                  DEFAULT              NORMAL             NO
 AP2                  DEFAULT              NORMAL             NO
 ASRT                 DEF2                 NORMAL             NO

 BRS [i 1] [dlci 16] Config>exit

BRS [i 1] Config>circuit 17
 BRS [i 1] [dlci 17] Config>list

 BANDWIDTH RESERVATION listing from SRAM
 bandwidth reservation is enabled
 interface number 1, circuit number 17 using defaults.
 maximum queue length 10, minimum queue length 3
 total bandwidth allocated 60%
 total classes defined (counting one local and one default) 4


 class LOCAL has 10% bandwidth allocated
   protocols and filters cannot be assigned to this class.


 class DEFAULT has 40% bandwidth allocated
   the following protocols and filters are assigned:
     protocol ARP with default priority is not discard eligible
     protocol DNA with default priority is not discard eligible
     protocol VINES with default priority is not discard eligible
     protocol IPX with default priority is not discard eligible
     protocol OSI with default priority is not discard eligible
     protocol AP2 with default priority is not discard eligible

 class DEF1 has 5% bandwidth allocated
   the following protocols and filters are assigned:
     protocol IP with priority NORMAL is not discard eligible


 class DEF2 has 5% bandwidth allocated
   the following protocols and filters are assigned:
     protocol ASRT with priority NORMAL is not discard eligible

 assigned tags:
```

```
default class is DEFAULT with priority NORMAL

BRS [i 1] [dlci 17] Config>add-class
This circuit is currently using circuit defaults...
Are you sure you want to override the defaults ?(Yes or [No]): yes
Class name [DEFAULT]? CIRC171
Percent bandwidth to reserve [10]? 5
BRS[i 1] [dlci 17] Config>assign vines
Class name [DEFAULT]? CIRC171
Priority <URGENT/HIGH/NORMAL/LOW> [NORMAL]?
Frame Relay Discard Eligible <NO/YES>[NO]?


BRS [i 1] [dlci 17] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, circuit number 17
maximum queue length 10, minimum queue length 3
total bandwidth allocated 65%
total classes defined (counting one local and one default) 5


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ARP with default priority is not discard eligible
    protocol DNA with default priority is not discard eligible
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible


class DEF1 has 5% bandwidth allocated
  the following protocols and filters assigned:
    protocol IP with priority NORMAL is not discard eligible


class DEF2 has 5% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ASRT with priority NORMAL is not discard eligible


class CIRC171 has 5% bandwidth allocated
  the following protocols and filters are assigned:
    protocol VINES with priority NORMAL is not discard eligible

assigned tags:

default class is DEFAULT with priority NORMAL

BRS [i 1] [dlci 17] Config>show

BANDWIDTH RESERVATION currently in RAM
interface number 1, circuit number 17
maximum queue length 10, minimum queue length 3
5 current defined classes:
  class LOCAL has 10% bandwidth allocated
  class DEFAULT has 40% bandwidth allocated
  class DEF1 has 5% bandwidth allocated
  class DEF2 has 5% bandwidth allocated
  class CIRC171 has 5% bandwidth allocated


protocol and filter assignments:

Protocol/Filter      Class             Priority    Discard Eligible
---------------      -----             --------    ----------------
IP                   DEF1                NORMAL            NO
ARP                  DEFAULT             NORMAL            NO
DNA                  DEFAULT             NORMAL            NO
VINES                CIRC171             NORMAL            NO
IPX                  DEFAULT             NORMAL            NO
OSI                  DEFAULT             NORMAL            NO
AP2                  DEFAULT             NORMAL            NO
ASRT                 DEF2                NORMAL            NO
```

```
BRS [i 1] [dlci 17] Config>exit
BRS [i 1] Config>set-circuit-defaults
BRS [i 1] [circuit defaults] Config>change DEF1
Percent bandwidth to reserve [ 5]? 10
BRS [i 1] [circuit defaults] Config>change DEF2
Percent bandwidth to reserve [5]? 10
BRS [i 1] [circuit defaults] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, default circuit
total bandwidth allocated 70%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ARP with default priority is not discard eligible
    protocol DNA with default priority is not discard eligible
    protocol VINES with default priority is not discard eligible
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible

class DEF1 has 10% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL is not discard eligible


class DEF2 has 10% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ASRT with priority NORMAL is not discard eligible

assigned tags:

default class is DEFAULT with priority NORMAL

BRS [i 1] [circuit defaults] Config>exit


BRS [i 1] Config>circuit 16
  BRS [i 1] [dlci 16] Config>list

  BANDWIDTH RESERVATION listing from SRAM
  bandwidth reservation is enabled
  interface number 1, circuit number 16 using defaults.
  maximum queue length 10, minimum queue length 3
  total bandwidth allocated 70%
  total classes defined (counting one local and one default) 4


  class LOCAL has 10% bandwidth allocated
    protocols and filters cannot be assigned to this class.


  class DEFAULT has 40% bandwidth allocated
    the following protocols and filters are assigned:
      protocol ARP with default priority is not discard eligible
      protocol DNA with default priority is not discard eligible
      protocol VINES with default priority is not discard eligible
      protocol IPX with default priority is not discard eligible
      protocol OSI with default priority is not discard eligible
      protocol AP2 with default priority is not discard eligible

  class DEF1 has 10% bandwidth allocated
    the following protocols and filters are assigned:
      protocol IP with priority NORMAL is not discard eligible


  class DEF2 has 10% bandwidth allocated
    the following protocols and filters are assigned:
      protocol ASRT with priority NORMAL is not discard eligible

  assigned tags:
```

```
  default class is DEFAULT with priority NORMAL

  BRS [i 1] [dlci 16] Config>exit


BRS [i 1] Config>circuit 17
  BRS [i 1[ [dlci 17] Config>list

  BANDWIDTH RESERVATION listing from SRAM
  bandwidth reservation is enabled
  interface number 1, circuit number 17
  maximum queue length 10, minimum queue length 3
  total bandwidth allocated 65%
  total classes defined (counting one local and one default) 5


  class LOCAL has 10% bandwidth allocated
    protocols and filters cannot be assigned to this class.


  class DEFAULT has 40% bandwidth allocated
    the following protocols and filters are assigned:
      protocol ARP with default priority is not discard eligible
      protocol DNA with default priority is not discard eligible
      protocol IPX with default priority is not discard eligible
      protocol OSI with default priority is not discard eligible
      protocol AP2 with default priority is not discard eligible


  class DEF1 has 5% bandwidth allocated
    the following protocols and filters are assigned:
      protocol IP with priority NORMAL is not discard eligible


  class DEF2 has 5% bandwidth allocated
    the following protocols and filters are assigned:
      protocol ASRT with priority NORMAL is not discard eligible


  class CIRC171 has 5% bandwidth allocated
    the following protocols and filters are assigned:
      protocol VINES with priority NORMAL is not discard eligible

  assigned tags:

  default class is DEFAULT with priority NORMAL

  BRS [i 1] [dlci 17] Config>use-circuit-defaults    9
  This circuit is currently NOT using circuit defaults...
  Are you sure you want to delete current definitions and use defaults ? (Yes or
  [No]): yes
  Defaults are in effect for this circuit.
  Please reload router for this command to take effect.
  BRS [i 1] [dlci 17] Config>
  *restart
  Are you sure you want to reload the gateway? (Yes or [No] ):yes


  *t 6
  Gateway user configuration
  Config>feature brs
  Bandwidth Reservation User Configuration
  BRS Config>interface 1
  BRS [i 1] Config>circuit 17
  BRS [i 1] [dlci 17] Config>list

  BANDWIDTH RESERVATION listing from SRAM
  bandwidth reservation is enabled
  interface number 1, circuit number 17 using defaults.
  maximum queue length 10, minimum queue length 3
  total bandwidth allocated 70%
  total classes defined (counting one local and one default) 4


  class LOCAL has 10% bandwidth allocated
    protocols and filters cannot be assigned to this class.


   class DEFAULT has 40% bandwidth allocated
    the following protocols and filters are assigned:
      protocol ARP with default priority is not discard eligible
```

```
    protocol DNA with default priority is not discard eligible
    protocol VINES with default priority is not discard eligible
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible

class DEF1 has 10% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL is not discard eligible


class DEF2 has 10% bandwidth allocated
  the following protocols and filters are assigned:
    protocol ASRT with priority NORMAL is not discard eligible

assigned tags:

default class is DEFAULT with priority NORMAL


BRS [i 1] [dlci 17] Config>show

BANDWIDTH RESERVATION currently in RAM
interface number 1, circuit number 17 using defaults.
maximum queue length 10, minimum queue length 3
4 current defined classes:
  class LOCAL has 10% bandwidth allocated
  class DEFAULT has 40% bandwidth allocated
  class DEF1 has 10% bandwidth allocated
  class DEF2 has 10% bandwidth allocated

protocol and filter assignments:

Protocol/Filter      Class              Priority    Discard Eligible
---------------      -----              --------    ----------------
IP                   DEF1                 NORMAL            NO
ARP                  DEFAULT              NORMAL            NO
DNA                  DEFAULT              NORMAL            NO
VINES                DEFAULT              NORMAL            NO
IPX                  DEFAULT              NORMAL            NO
OSI                  DEFAULT              NORMAL            NO
AP2                  DEFAULT              NORMAL            NO
ASRT                 DEF2                 NORMAL            NO

BRS [i 1] [dlci 17] Config>exit
```

# Chapter 58. Configuring and Monitoring Bandwidth Reservation

This chapter describes the Bandwidth Reservation System (BRS) configuration and operational commands.

This chapter includes the following sections:



## Bandwidth Reservation Configuration Overview

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. Reload your router.
6. Repeat steps 1 through 3 to configure bandwidth reservation for the particular interface that you have enabled.
7. If you are configuring BRS on a PPP interface, at the `BRS[i 0]Config>` prompt, configure traffic classes and assign protocols, filters, and tags to the traffic classes using the configuration commands listed in Table 100 on page 702. If you are configuring BRS on a FR interface, follow steps 8 through 10.
8. If you are configuring BRS on a FR interface, you can configure circuit classes and assign circuits to circuit classes using the commands listed in Table 99 on page 701
9. If you want to use default circuit definitions then enter the **set-circuit-defaults** command at the `BRS[i 0]Config>` prompt. This gets you to the `BRS[i 0][circuit defaults]` prompt where you can use the appropriate commands from Table 100 on page 702 to configure traffic classes and assign protocols,

filters, and tags to the traffic classes. Once you are through defining default circuit definitions for traffic class handling, enter ″exit″ to return to the `BRS[i 0] Config>` prompt.

10. If you have FR circuits that cannot use default circuit definitions for traffic class handling, enter **circuit** *permanent-virtual-circuit circuit_number*. This will access the circuit prompt where you can use the commands listed in Table 100 on page 702 to create circuit-specific definitions for traffic class handling.

**Note:** You do not need to reload 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 699, 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 4.b on page 699 , 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 reload 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>`, or `BRS [i x] [circuit defaults] Config>`).

*Table 98. Bandwidth Reservation Configuration Command Summary (Available from BRS Config> prompt)*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |

*Table 98. Bandwidth Reservation Configuration Command Summary (Available from BRS Config> prompt) (continued)*

| Command | Function |
|---|---|
| Activate-IP-precedence-filtering | Activate BRS IPv4 precedence filtering of APPN and SNA packets that are sent over a secure IP tunnel or that are in secondary TCP or UDP fragments. You also must configure the setting of the IPv4 precedence bits when you configure DLSw, HPR over IP or TN3270. |
| Deactivate-IP-precedence-filtering | Deactivates IPv4 precedence filtering processing. |
| Enable-hpr-over-ip-port-numbers | Enables the use of BRS filtering for APPN-HPR over IP traffic and allows the configuration of the UDP port numbers used to identify HPR over IP packets. **Note:** If APPN is in the load image, this command is not supported since BRS learns from APPN if HPR over IP has been configured and, if it has been configured, learns the UDP port numbers that will be used for HPR over IP packets from the APPN support. |
| Disable-hpr-over-ip-port-numbers | Disables BRS filtering of APPN-HPR over IP traffic. **Note:** If APPN is in the load image, this command is not supported since BRS learns from APPN whether or not HPR over IP has been configured. |
| Interface | Selects an interface on which to configure bandwidth reservation. **Note:** This command must be entered before using any other configuration commands. See Table 99 and Table 100 on page 702. |
| List | Lists the interfaces that can support bandwidth reservation and, for each interface, indicates if bandwidth reservation is enabled or disabled. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

*Table 99. BRS Interface Configuration Commands Available from BRS [i #] Config> prompt for Frame Relay Interfaces*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add-circuit-class | Sets the name of a bandwidth c-class and its percentage of bandwidth. |
| Assign-circuit | Assigns a specified circuit to the specified bandwidth c-class. |
| Change-circuit-class | Changes the amount of bandwidth configured for a bandwidth c-class. |

*Table 99. BRS Interface Configuration Commands Available from BRS [i #] Config> prompt for Frame Relay Interfaces (continued)*

| Command | Function |
|---|---|
| Circuit | Accesses the BRS circuit-level prompt (`BRS [i x][dlci y] Config>`) prompt where you can use the commands listed in Table 100 to configure Bandwidth Reservation on the Frame Relay circuit. |
| Clear-block | Clears the configuration data associated with the current interface from SRAM. Circuit class configuration data and default circuit definitions for traffic class handling are cleared. |
| Deassign-circuit | Restores the specified circuit to the default c-class |
| Default-circuit-class | Assigns the name of a default bandwidth c-class and its percentage of the interface's bandwidth. |
| Del-circuit-class | Deletes the specified bandwidth c-class. |
| Disable | Disables bandwidth reservation on the interface . |
| Enable | Enables bandwidth reservation on the interface. |
| List | Displays the c-classes and assigned circuit definitions from SRAM. |
| Queue-length | Sets the maximum and minimum values for the number of packets in a priority queue. |
| Set-circuit-defaults | Accesses the `BRS [i x] [circuit defaults] Config>` command prompt so that you can use the appropriate commands from Table 100 to create default circuit definitions for traffic class handling. |
| Show | Displays the currently defined c-classes and assigned circuits from SRAM. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

The following table lists BRS circuit commands Available from `BRS [i x] Config>` for PPP interfaces, `BRS [i x] dlci [y] Config>` prompt for Frame Relay circuits, and from the `BRS [i x] [circuit defaults] Config>` prompt.

*Table 100. BRS Traffic Class Handling Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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. |
| Change-class | Changes the amount of bandwidth configured for a bandwidth t-class. |
| Clear-block | Clears the traffic class and protocol, filter, and tag assignment configuration data from SRAM for the PPP interface or Frame Relay circuit.<br>**Note:** This command cannot be used from the `BRS [i x] [circuit defaults] Config>` prompt. |
| Deassign | Restores the queuing of the specified packet or filter to the default t-class and priority. |
| Default-class | Sets the default t-class and priority to a desired value and assigns all unassigned protocols to the new default t-class. |
| Del-class | Deletes a previously configured bandwidth t-class. |

*Table 100. BRS Traffic Class Handling Commands (continued)*

| Command | Function |
|---|---|
| Disable | Disables bandwidth reservation on the PPP interface or Frame Relay circuit.<br>**Note:** BRS cannot be enabled or disabled from the `BRS [i x] [circuit defaults] Config>` prompt. |
| Enable | Enables bandwidth reservation on the PPP interface or Frame Relay circuit.<br>**Note:** BRS cannot be enabled or disabled from the `BRS [i x] [circuit defaults] Config>` prompt. |
| List | Lists the configured t-classes and protocol, filter and tag assignments stored in SRAM. |
| Queue-length | Sets the maximum and minimum values for the number of packets in a priority queue.<br>**Note:** This command is not supported at the `BRS [i x] [circuit defaults] Config>` prompt. |
| Show | Displays the currently defined t-classes and protocol, filter, and tag assignments stored in RAM.<br>**Note:** This command is not supported at the `BRS [i x] [circuit defaults] Config>` prompt. |
| Tag | Assigns a BRS tag name (TAG1 - TAG5) to a MAC 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 MAC filter that has been tagged during configuration of the MAC filtering feature. |
| Use-circuit-defaults | Allows the user to delete the circuit-specific definitions and use the circuit-defaults definitions for the traffic-class handling. This command is valid at the `BRS [i x] dlci [y] Config>` prompt for Frame Relay only.<br>**Note:** The router must be reloaded in order for the defaults to become operational. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

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 reload before taking effect are the enable, disable, use-circuit-defaults, and clear-block commands.

3. You do not need to reload the router for t-class and c-class configuration changes to take effect.

## Activate-IP-precedence-filtering

Use the **activate-ip-precedence-filtering** command to activate BRS IPv4 precedence filtering of APPN and SNA packets that are sent over a secure IP tunnel or that are in secondary TCP or UDP fragments. You also must configure the setting of the IPv4 precedence bits when you configure DLSw, HPR over IP or TN3270. See "Using IP Version 4 Precedence Bit Processing for SNA Traffic in IP Secure Tunnels and Secondary Fragments" on page 687 for more information.

**Syntax:**

**activate-ip-precedence-filtering**

## Add-circuit-class

**Note:** 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:**

**add-circuit-class** *class-name %*

## Add-class

Use the **add-class** command to allocate a designated amount of bandwidth to a user-defined bandwidth t-class.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer "Yes", the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer "No", the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the `BRS [i x][circuit defaults]Config>` command prompt.

**Syntax:**

**add-class** [*class-name* or *class#*] *%*

**Example:**

```
BRS [i 1] [dlci 17] Config>add-class
This circuit is currently using circuit defaults...
Are you sure you want to override the defaults ?(Yes or [No]):y
Class name [DEFAULT]? CIRC17
Percent bandwidth to reserve [10]?5
BRS [i 1] [dlci 17] Config>list
```

```
BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, circuit number 17
maximum queue length 10, minimum queue length 3
total bandwidth allocated 65%
total classes defined (counting one local and one default) 5


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol DNA with default priority is not discard eligible
    protocol VINES with default priority is not discard eligible
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible
    protocol ASRT with default priority is not discard eligible


class DEF1 has 5% bandwidth allocated
  protocol IP with priority NORMAL is not discard eligible.


class DEF2 has 5% bandwidth allocated
 protocol ARP with priority NORMAL is not discard eligible.


class CIRC171 has 5% bandwidth allocated
  no protocols or filters are assigned to this class.

assigned tags:

default class is DEFAULT with priority NORMAL
```

## 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:**

**assign** [*protocol-class* or TAG or *filter-class*] [*class-name* or *class#*]

The **assign** command also allows you to set the Discard-eligible (DE) bit for Frame Relay frames.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer “Yes”, the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer “No”, the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the `BRS [i x][circuit defaults]Config>` command prompt.

**Example:**

```
assign IPX test
priority <URGENT/HIGH/NORMAL/LOW>: [NORMAL]? low
protocol IPX maps to class test with priority LOW Discard eligible <yes/no> [N]?
```

## Assign-circuit

**Note:** 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 reload the router before you can use this command to assign the circuit to a circuit class.

**Syntax:**

**assign-circuit** *# class name*

## Change-circuit-class

**Note:** 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:**

**change-circuit-class** *class-name %*

## Change-class

Use the **change-class** command to change the amount of bandwidth configured for a bandwidth t-class.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer “Yes”, the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer “No”, the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the `BRS [i x][circuit defaults]Config>` command prompt.

**Syntax:**

**change-class** [*class-name* or *class#*] *%*

## Circuit

**Note:** 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:**

***ci*rcuit** *permanent-virtual-circuit-#*

Before you can use the **add-class**, **assign**, **default-class**, **del-class**, **deassign**, or **change-class** commands, you must enable BRS on the circuit and reload 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 reloaded, 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 | use-circuit-defaults | |

## Clear-block

Use the **clear-block** command to clear the current bandwidth reservation configuration data from SRAM.

**Syntax:**

**clear-block**

- 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 and default circuit definitions for traffic class handling are 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 reloaded.

**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>
```

## Deactivate-IP-precedence-filtering

Use the **deactivate-ip-precedence-filtering** command to deactivate IPv4 precedence filtering processing.

**Syntax:**

**deactivate-ip-precedence-filtering**

## Deassign

Use the **deassign** command to restore the queuing of the specified protocol packet or filter to the default t-class and priority.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer "Yes", the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer "No", the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the `BRS [i x][circuit defaults]Config>` command prompt.

**Syntax:**

**deassign** [*prot-class* or *filter-class*]

## Deassign-circuit

**Note:** 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-c** *#*

## Default-circuit-class

**Note:** 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:**

**default-circuit-class** *class-name %*

## Del-circuit-class

**Note:** Used only when configuring Frame Relay.

Use the **del-circuit-class** command at the interface level to delete the specified bandwidth c-class.

**Syntax:**

**del-circuit-class** *class-name*

## 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.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer "Yes", the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer "No", the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the `BRS [i x][circuit defaults]Config>` command prompt.

**Syntax:**

**default-cl** [*class-name* or *class#*] *priority*

## Del-class

Use the **del-class** command to delete a previously configured bandwidth t-class from the specified interface or circuit.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer "Yes", the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer "No", the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the `BRS [i x][circuit defaults]Config>` command prompt.

**Syntax:**

**del-class** [*class-name* or *class#*]

## 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 reloaded.

To verify that bandwidth reservation is disabled, enter the **list** command.

**Syntax:**

**disable**

## Disable-hpr-over-ip-port-numbers

Use the **disable-hpr-over-ip-port-numbers** command to disable BRS filtering of HPR over IP traffic.

**Syntax:**

**disable-hpr-over-ip-port-numbers**

To verify that BRS filtering of HPR over IP traffic is disabled, enter the **list** command.

**Note:** If APPN is included in the load image, you configure whether or not HPR over IP traffic will be used at the `APPN Config>` command prompt.

## 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 reloaded.

**Syntax:**

**enable**

**Note:**

- When configuring BRS on a PPP interface, issue the **enable** command at the interface prompt, and then reload the router before configuring any traffic classes and assigning protocols and filters to traffic classes.
- When BRS is initially enabled on a Frame Relay circuit, the circuit is initialized to use default circuit definitions for traffic class handling. 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 reload the router before configuring circuit classes for the interface and traffic classes for each circuit. For example:

```
t 6
Gateway user configuration
Config>f brs
Bandwidth Reservation User Configuration
BRS Config>interface 1
BRS [i 1] Config>enable
Please reload  router for this command to take effect
BRS [i 1] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1
maximum queue length 10, minimum queue length 3
total bandwidth allocated 10%
total circuit classes defined (counting one default) 1


class DEFAULT has 10% bandwidth allocated
  no circuits are assigned to this class.

default class is DEFAULT

BRS [i 1] Config>circ 16
BRS [i 1] [dlci 16] Config>enable
Defaults are in effect for this circuit.
Please reload  router for this command to take effect.
BRS [i 1] [dlci 16] Config>ex
Please reload  router for this command to take effect.
BRS [i 1] [dlci 16] Config>
```

## Enable-hpr-over-ip-port-numbers

Use the **enable-hpr-over-ip-port-numbers** command to enable BRS filtering of APPN-HPR over IP traffic and to configure UDP port numbers used to identify HPR over IP packets.

**Note:** If APPN is included in the load image, you enable HPR over IP and specify the UDP port numbers used for HPR over IP traffic at the `APPN Config>` command prompt.

**Syntax:**

**enable-hpr-over-ip-port-numbers**

**Example:**

```
BRS Config> enable-hpr-over-ip-port-numbers
XID exchange port number [12000]?
HPR net  trans prio port number [12001]?
HPR high trans prio port number [12002]?
HPR medium trans prio port number [12003]?
HPR low    trans prio port number [12004]?
```

**XID exchange port number**

This parameter specifies the UDP port number to be used for XID exchange. This port number must be the same as the one defined on other devices in the network.

**Valid Values**: 1024 - 65535

**Default Value**: 12000

**Network priority port number**

This parameter specifies the UDP port number to be used for network priority traffic. This port number must be the same as the one defined on other devices in the network.

**Valid Values**: 1024 - 65535

**Default Value**:12001

**High exchange port number**

This parameter specifies the UDP port number to be used for high priority traffic. This port number must be the same as the one defined on other devices in the network.

**Valid Values**: 1024 - 65535

**Default Value**:12002

**Medium exchange port number**

This parameter specifies the UDP port number to be used for medium priority traffic. This port number must be the same as the one defined on other devices in the network.

**Valid Values**: 1024 - 65535

**Default Value**:12003

**Low exchange port number**

This parameter specifies the UDP port number to be used for low priority traffic. This port number must be the same as the one defined on other devices in the network.

**Valid Values**: 1024 - 65535

**Default Value**:12004

## 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.*

**Syntax:**

**interface** *interface#*

**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 reload 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 reload the router.

## List

Use the **list** command to display currently defined bandwidth classes and their guaranteed percentage rates.

The **list** command and **show** command are similar. The **list** command displays the current SRAM definitions and the **show** command displays the current RAM definitions.

**Syntax:**

**list** *interface#*

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 1] [dlci 16] Config>`
- `BRS [i 1] Config>`
- `BRS Config>`
- `BRS [i 1] [circuit defaults] Config>`

**Note:** When you use this command from a Frame Relay circuit prompt (`BRS [i x] [dlci y] Config>`) it indicates if the circuit is using default circuit definitions or circuit-specific definitions for traffic class handling. If the circuit is using default circuit definitions, the traffic class, protocol, filter, and tag assignments currently defined for default circuit definitions are displayed. However, if you want to alter the default circuit definitions, you need to get to the `BRS[i x] [circuit defaults] Config>` prompt to make changes.

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.

### Example 1

```
BRS Config>list
 Bandwidth Reservation is available for 2 interfaces.

 Interface   Type            State
 ---------   ----            -----
         1   FR              Enabled
         2   PPP             Enabled

 The use of HPR over IP port numbers is disabled

 BRS Config>interface 1
 BRS [i 1] Config>list

 BANDWIDTH RESERVATION listing from SRAM
 bandwidth reservation is enabled
 interface number 1
 maximum queue length 10, minimum queue length 3
 total bandwidth allocated 10%
 total circuit classes defined (counting one default) 1


 class DEFAULT has 10% bandwidth allocated
   the following circuits are assigned:
        17
        16  using defaults.
        18  using defaults.

 default class is DEFAULT

 BRS [i 2] Config>exit
 BRS Config>interface 2
 BRS [i 2] Config>list

 BANDWIDTH RESERVATION listing from SRAM
 bandwidth reservation is enabled
 interface number 2
 maximum queue length 10, minimum queue length 3
 total bandwidth allocated 50%
 total classes defined (counting one local and one default) 2


 class LOCAL has 10% bandwidth allocated
   protocols and filters cannot be assigned to this class.


 class DEFAULT has 40% bandwidth allocated
   the following protocols and filters are assigned:
     protocol IP with default priority
     protocol ARP with default priority
     protocol DNA with default priority
     protocol VINES with default priority
     protocol IPX with default priority
     protocol OSI with default priority
     protocol AP2 with default priority
     protocol ASRT with default priority

 assigned tags:
```

```
default class is DEFAULT with priority NORMAL

BRS [i 2] Config>
```

### Example 2

```
BRS [i 1] [dlci 17] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
maximum queue length 10, minimum queue length 3
total bandwidth allocated 60%
total classes defined (counting one local and one default) 3


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
   protocol ASRT with priority NORMAL is not discard eligible
   filter NETBIOS with priority NORMAL is not discard eligible


class CLASS1 has 10% bandwidth allocated
  the following protocols and filters are assigned:
   protocol IP with priority NORMAL is not discard eligible
   protocol ARP with priority NORMAL is not discard eligible
   protocol DNA with priority NORMAL is not discard eligible
   protocol VINES with priority NORMAL is not discard eligible
   protocol IPX with priority NORMAL is discard eligible
   protocol OSI with priority NORMAL is not discard eligible
   protocol AP2 with priority NORMAL is not discard eligible
```

### Example 3

```
BRS [i 1] [circuit defaults] Config>list

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 1, default circuit
maximum queue length 10, minimum queue length 3
total bandwidth allocated 70%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.



class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol DNA with default priority is not discard eligible
    protocol VINES with default priority is not discard eligible
    protocol IPX with default priority is not discard eligible
    protocol OSI with default priority is not discard eligible
    protocol AP2 with default priority is not discard eligible
    protocol ASRT with default priority is not discard eligible


class DEF1 has 10% bandwidth allocated
  protocol IP with priority NORMAL is not discard eligible.


class DEF2 has 10% bandwidth allocated
 protocol ARP with priority NORMAL is not discard eligible.

assigned tags:

default class is DEFAULT with priority NORMAL

 BRS [i 1] [circuit defaults] Config>
```

### Example 4

```
BRS Config>list
 Bandwidth Reservation is available for 2 interfaces.

 Interface   Type            State
 ---------   ----            -----
         1   FR              Enabled
         2   PPP             Enabled

 The use of HPR over IP port numbers is enabled.
```

```
Transmission Type   Port Number
-----------------   -----------
    XID exchange        12000
    HPR network         12001
    HPR high            12002
    HPR medium          12003
    HPR low             12004
```

## 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.

**Syntax:**

**queue-length** *maximum-length minimum-length*

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, 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.

## Set-circuit-defaults

Use the **set-circuit-defaults** command to access the commands used to define default circuit definitions for traffic class handling. These default circuit definitions can then be used by any Frame Relay circuits on the interface that can use the same traffic classes and protocol, filter, and tag assignments.

**Syntax:**

**set-circuit-defaults**

## Show

Use the **show** command to display currently defined bandwidth classes stored in RAM.

**Syntax:**

**show** *interface#*

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*. The following example shows the output of the show command from the circuit level prompt.

```
BRS [i 1] [dlci 17] Config>show

Protocol/Filter      Class              Priority    Discard Eligible
---------------      -----              --------    ----------------
IP                   CLASS1               NORMAL             NO
ARP                  CLASS1               NORMAL             NO
DNA                  CLASS1               NORMAL             NO
VINES                CLASS1               NORMAL             NO
IPX                  CLASS1               NORMAL             YES
OSI                  CLASS1               NORMAL             NO
AP2                  CLASS1               NORMAL             NO
ASRT                 DEFAULT              NORMAL             NO
NETBIOS              DEFAULT              NORMAL             NO
```

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.

**Notes:**

1. When you use this command from a Frame Relay circuit prompt (`BRS [i x] [dlci y] Config>`) it indicates if the circuit is using default circuit definitions or circuit-specific definitions for traffic class handling. If the circuit is using default circuit definitions, the traffic class, protocol, filter, and tag assignments currently defined for default circuit definitions are displayed. However, if you want to alter the default circuit definitions, you need to get to the `BRS [i x] [circuit defaults] Config>` prompt to make changes.
2. This command cannot be used from the `BRS [i x] [circuit defaults] Config>` prompt.

# Tag

Use the **tag** command to assign a MAC filter item 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.

**Syntax:**

**tag** *mac_filter_tag#*

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.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer “Yes”, the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer “No,” the command is aborted and default circuit definitions will continue to be used

for the circuit. If you want to change the default circuit definitions, you should go to the BRS [i x][circuit defaults]Config> command prompt.

## 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.

**Syntax:**

**untag** *mac_filter_tag#*

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.

**Note:** If this command is used for a Frame Relay circuit that is currently using default circuit definitions for traffic class handling, you will be asked whether or not you want to override the default circuit definitions. If you answer “Yes”, the circuit will be changed to use circuit-specific definitions for traffic class handling and the command will be allowed. If you answer “No”, the command is aborted and default circuit definitions will continue to be used for the circuit. If you want to change the default circuit definitions, you should go to the BRS [i x][circuit defaults]Config> command prompt.

## Use-circuit-defaults

Use the **use-circuit-defaults** command at the circuit level to delete the circuit-specific definitions and use the circuit default definitions for traffic-class handling. You will be prompted to confirm that you want to use the circuit defaults.

**Syntax:**

**use-circuit-defaults**

**Notes:**

1. This command is used only when configuring Frame Relay
2. The router must be reloaded for the defaults to become operational.

**Example:**

```
BRS [i 1] [dlci 17] Config>use-circuit-defaults
This circuit is currently NOT using circuit defaults...
 Are you sure you want to delete current definitions and use defaults ? (Yes or
 [No]): y
 Defaults are in effect for this circuit.
 Please reload  router for this command to take effect.
 BRS [i 1] [dlci 17] Config>
```

# Accessing the Bandwidth Reservation Monitoring Prompt

To access bandwidth reservation monitoring 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 Monitoring Commands".)

   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 monitoring prompt (`BRS>`), you can enter any of the specific monitoring commands described in Table 101.

## Bandwidth Reservation Monitoring Commands

This section summarizes and explains the Bandwidth Reservation monitoring commands. 101 shows the Bandwidth Reservation monitoring commands. The commands that can be used differ depending on the BRS monitoring prompt (`BRS>`, `BRS [i x]>`, or `BRS [i x] [dlci y]>`).

*Table 101. Bandwidth Reservation Monitoring Command Summary*

| Command | Used Only With FR | Function |
|---|---|---|
| ? (Help) | | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10 |
| 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.<br>**Note:** This command must be entered before using any other bandwidth reservation monitoring commands. |
| Last | | Displays the last saved t-class counters. |
| Last-circuit-class | yes | Displays the last saved c-class counters. |

*Table 101. Bandwidth Reservation Monitoring Command Summary (continued)*

| Command | Used Only With FR | Function |
|---|---|---|
| Exit | | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11 |

## Circuit

**Note:** 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:**

**circuit** *permanent-virtual-circuit-#*

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:**

**clear**

## Clear-Circuit-Class

**Note:** 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:**

**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:**

**counters**

**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
```

**Note:** The Bytes Ovfl column lists the number of bytes for packets that could not be transmitted because either the maximum queue-length was reached for a priority queue or the packet could not be queued because the priority queue was at the minimum queue length threshold and the packet came from an interface that was running low on receive buffers.

## Counters-Circuit-Class

**Note:** 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:**

**counters-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 monitoring commands will be applied. *Bandwidth reservation is supported on routers running the PPP (Point-to-Point Protocol) and Frame Relay interfaces.*

**Syntax:**

**interface** *interface#*

**Note:** To enter bandwidth reservation commands for a new interface, this command must be entered before using any other bandwidth reservation monitoring commands. If you have exited the bandwidth reservation monitoring 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> monitoring prompt, type the number of the interface. You can then use bandwidth reservation monitoring commands as described in this chapter.

## 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**

## Last-Circuit-Class

**Note:** 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**

# Chapter 59. Using 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:



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 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 are marked with a number in the range 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.

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 INCLUDE, EXCLUDE, or TAG. In addition, if the default action is TAG, a tag value is also given.

## Using MAC Filtering Tags

The following list includes some uses of 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 to 64.
- 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 (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.

# Chapter 60. Configuring and Monitoring MAC Filtering

This chapter describes how to access the MAC Filtering configuration and monitoring prompts and how to use the available commands. It includes the following sections:



## 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 the MAC filtering configuration commands. Enter these commands at the `Filter config>` prompt.

Use the following commands to configure the MAC filtering feature.

*Table 102. MAC Filtering Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| 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 | Removes a filter list from a filter. |
| Disable | Disables MAC Filtering entirely or disables a particular filter. |

*Table 102. MAC Filtering Configuration Command Summary (continued)*

| Command | Function |
|---|---|
| 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 | Re-initializes the entire MAC Filtering system from an updated 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## 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:**

**attach** *filter-list-name filter-number*

## Create

Use the **create** command to create a filter-list or an INPUT or OUTPUT filter.

**Syntax:**

**create** list *filter-list-name*

filter [input or 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.

**filter [input or 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, has a default action of INCLUDE and is ENABLED.

## 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.

**include** *filter-number*
Sets the default action for the filter with a specified filter number to include.

**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.

## 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.

**filter** *filter-number*
Deletes a filter that was created using the **create filter** command.

## Detach

Use the **detach** command to delete a filter-list name (filter-list parameter) from a filter (filter-number parameter).

**Syntax:**

**detach** *filter-list-name filter-number*

## 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.

**filter** *filter-number*
Disables a particular filter. The filter-number parameter corresponds to the numbers displayed in the **list filters** command.

## 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.

**filter** *filter-number*
Enables a particular filter. The filter-number parameter corresponds to the numbers displayed in the **list filters** command.

## 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*

**all** Displays a summary of all the configured filter-lists and filters.

**filter** *filter-number*
Generates a list of attached filter-lists for the specified filter and all subsequent information for the filter.

## 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*

## Reinit

Use the **reinit** command to re-initialize the entire MAC Filtering system from an updated configuration, without affecting the rest of the router.

**Syntax:**

**reinit**

## Set-Cache

Use the **set-cache** command to change the default cache size (16) to a number in the range 4 to 32768.

**Syntax:**

**set-cache** *cache-size filter-number*

## 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*

# Update Subcommands

This section summarizes the MAC filtering configuration subcommands. Enter these subcommands at the `Filter filter-list-name config>` prompt.

*Table 103. Update Subcommands Summary*

| Subcommand | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

Use the following subcommands to update a filter-list.

## 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, or a sequence of window data with a mask to compare against a packet data.

The order in which the filter-items are added to a given filter-list is important because 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, and so on. After you enter a successful **add** subcommand, 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 INCLUDE, EXCLUDE, or TAG, depending on the designated action of the filter-list. If none of the filter-items of a filter-list produces a match, then the default action (INCLUDE, EXCLUDE or TAG) of the filter is returned.

**Syntax: add** 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 to compare against the source MAC address. ***hex-MAC-addr*** must be an even number of hex digits with a maximum of 16 digits and should be entered without a 0x in front.

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 (for example, 000003001234). It may also be represented as a series of hex digits with a hyphen (-) between every two digits (for example, 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 (for example, 00:00:C9:09:66:49). MAC addresses of filter-items will always be displayed using either a hyphen (-) or a colon (:) to distinguish canonical from noncanonical representations.

**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 the source MAC address of the packet.

**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) that matches the hex data with the mask against packet data.

**window INFO** *offset-value hex-data hex-mask*
Similar to the **add window mac** command, except that the offset is computed with respect to the beginning of the information field.

## 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:**

**delete** *filter-item-number*

## List

Use the **list** subcommand to print out a listing of all the 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

**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. It also gives the filter-list action.

**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.

**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. It also gives the filter-list action.

**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. It also gives the filter-list action.

**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. It also gives the filter-list action.

## 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*

## 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*

## Accessing the MAC Filtering Monitoring Prompt

Use the **feature** command from the GWCON process to access the MAC filtering monitoring commands. The **feature** command lets you access monitoring commands for specific router features outside of the protocol and network interface monitoring 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 monitoring prompt, enter the **feature** command followed by the feature number (3) or short name (MCF). For example:

```
+ feature mcf
MAC Filtering user monitoring
Filter>
```

Once you access the MAC filtering monitoring prompt, you can begin entering specific monitoring commands. To return to the GWCON prompt at any time, enter the **exit** command at the MAC Filtering monitoring prompt.

## MAC Filtering Monitoring Commands

This section summarizes the MAC filtering monitoring commands. Enter these commands at the `Filter>` prompt.

*Table 104. MAC Filtering Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| 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 | Re-initializes the entire MAC Filtering system from an updated configuration, without affecting the rest of the router. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

Use the following commands to monitor the MAC filtering feature.

## Clear

Use the **clear** command to clear filter statistics.

**Syntax:**

**clear** all

filter *filter-number*

**all** Clears the statistics listed by the **list all** command.

**filter** *filter-number*
Clears the statistics listed by the **list filter** command.

## 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.

**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.

## 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.

**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.

## 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.

**filter** *filter-number*
Generates statistics and settings for each filter plus all the filter-lists currently running in this filter.

## Reinit

Use the **reinit** command to re-initialize the entire MAC Filtering system from an updated configuration, without affecting the rest of the router.

**Syntax:**

**reinit**

## Configuring MAC Filtering

# Chapter 61. Using WAN Restoral

This chapter includes the following sections:



## Overview for WAN Restoral, WAN Reroute, and Dial-on-Overflow

The WAN Restoral, WAN Reroute, and Dial-on-overflow features have similar functions and might be confused. This overview is intended to help you decide which of these functions will be useful to you and to help you find the information you need to configure them.

The configuration commands for all three features are included in the ″Configuring WAN Restoral″ chapter. For additional information about WAN Reroute and Dial-on-overflow see "Chapter 63. The WAN Reroute Feature" on page 759.

## WAN Restoral

WAN Restoral is the most basic function. When you use WAN Restoral, you configure a primary and a secondary link. In case the primary link fails, the secondary link is started and assumes the characteristics of the primary. You don't configure any protocol definitions on the secondary link because it uses the protocol definitions from the primary link.

**For WAN Restoral:**

- There is a pairing between a primary and a secondary link.
- You can configure only one primary to use a specific secondary link.
- You don't configure protocol definitions (for example: protocol addresses) on the secondary link.
- The primary link must be a PPP serial interface, it can not be a PPP dial circuit interface.
- The secondary link must be a PPP dial circuit or a Multilink-PPP interface.
- You must enable the WRS feature using the **enable wrs** command.
- You must enable the primary/secondary pair using the **enable secondary-circuit** command.

**Note:** When BRS is configured on a primary link and the primary link is part of a primary-secondary pair for WAN Restoral, you must configure BRS on the secondary link. Typically when WAN Restoral is configured, the secondary link takes the identify of the primary link. However, this is not true for BRS; therefore, BRS needs to be configured on both the primary and secondary link.

## WAN Reroute

WAN Reroute is a more advanced function. When you use WAN Reroute, you configure a primary and an alternate link. In case the primary link fails, the alternate link is started. The routing protocols (for example, RIP or OSPF) detect the newly available link and adjust the routes that are used for forwarding packets.

**For WAN Reroute:**

- There is a pairing between a primary and an alternate link.
- You may configure multiple primary links to use the same alternate link.
- You must configure protocol definitions on the alternate link.
- The primary link may be any link on which you can configure routable protocols (e.g. IP, IPX). For example, the primary link may be a LAN interface, a PPP, Frame Relay, or X.25 serial interface, or a PPP or Frame Relay dial circuit. The following are examples of interface types that cannot be primary links: SDLC serial interfaces, SRLY serial interfaces, and base nets like V.25bis and ISDN.
- The alternate link may be any link on which you can configure routable protocols (e.g. IP, IPX) and the datalink type of the alternate link need not match the datalink type of the primary link. For example, the alternate link may be a LAN interface, a PPP, Frame Relay, or X.25 serial interface, or a PPP or Frame Relay dial circuit. The following are examples of interface types that cannot be alternate links: SDLC serial interfaces, SRLY serial interfaces, and base nets like V.25bis and ISDN.
- If the primary link is a dial circuit then it cannot be a dial-on-demand dial circuit (you must configure 'set idle 0' on the dial circuit). I.430, I.431 and Channelized T1/E1 Dial Circuits are implicitly fixed, and therefore can be used as a WRS Primary.

  **Note:** I.430/I.431 and Channelized T1/E1 dial circuits can be used as WRS primary without any explicit configuration.
- The alternate link may not be a dial-on-demand dial circuit (you must configure 'set idle 0' on the dial circuit).
- You must enable the WRS feature using the **enable wrs** command.
- You must enable the primary/alternate pair using the **enable alternate-circuit** command.
- You may optionally configure stabilization times and start-and stop-time-of-day-revert-back times to control the switching back to the primary link.
- If the alternate link is X.25, you should use the **national-personality set disconnect-procedure active** command when configuring the X.25 interface of the router that has WAN Reroute enabled and use the **national-personality set disconnect-procedure passive** command when configuring the X.25 interface of the other router.

## Dial-on-overflow

Dial-on-overflow is similar to WAN Reroute, but does not require failure of the primary to start the alternate link. Instead, the utilization of the primary link is monitored, and if a threshold is exceeded, the alternate link is started. Also, not all protocols are brought up on the alternate link. Only IP is brought up on the alternate link, and other protocols continue to use the primary link unless the primary link goes down.

If the primary link goes down, WAN Reroute takes over and any protocols configured on the alternate interface can start detecting and using routes on the alternate interface.

**For Dial-on-overflow:**

- Dial-on-overflow uses the primary/alternate pairing of a WAN Reroute pair.
- You must configure a WAN reroute pair to use Dial-on-overflow, and all the restrictions of WAN Reroute configuration apply.
- The primary link of a WAN Reroute pair that will be used for Dial-on-overflow must be Frame Relay.
- You must use the OSPF routing protocol to use Dial-on-overflow.
- You must use the **enable dial-on-overflow** command to configure add-threshold and drop-threshold, the bandwidth monitoring interval, and the minimum alternate up time.
- Stabilization times and start-time-of-day-revert-back and stop-time-of-day-revert-back times do not affect the operation of dial-on-overflow.

For more information about WAN Reroute see “Chapter 63. The WAN Reroute Feature” on page 759.

## Before You Begin

Before you configure WAN Restoral, you must have the following:

1. A primary serial interface (leased line) configured for PPP. You can use any serial interface on the router.
2. An interface with the associated dial circuits configured on the router. You can use an ISDN interface or a V.25bis interface as the base net.
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 Config>`prompt.
4. A secondary dial circuit at one 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. The protocol assignments for the primary interface are used on the secondary link (dial circuit) when it is active.
5. A secondary dial circuit at the other end of the link configured to receive calls only. Use the **set calls inbound** command at the `Circuit Config>` prompt.

## Configuration Procedure for WAN Restoral

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 PPP 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 datalink 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. Select the base net interface for the dial circuit. The base net can be V.25bis, or ISDN.

```
Circuit Config> set net 2
```

5. Set the dial circuit idle timer to 0 (0=fixed) as follows:

```
Circuit Config> set idle 0
```

6. Set one end of the backup connection to receive calls (for example, router A) as follows:

```
Circuit Config> set calls inbound
```

7. 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 dial circuit. The protocol assignments for the primary interface are used on the secondary interface (dial circuit) when it is active.
3. For ISDN configuration instructions, see “Chapter 51. Using the ISDN Interface” on page 629.
4. For V.25bis configuration instructions, see “Chapter 49. Using the V.25bis Network Interface” on page 613.

# Chapter 62. Configuring and Monitoring WAN Restoral

This chapter describes the WAN Restoral configuration and operational commands. It includes the following sections:



## WAN Restoral, WAN Reroute, and Dial-on-Overflow 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 105 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 105. WAN Restoral Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Add | Adds a mapping of primary-to-secondary (for WAN Restoral) or primary-to-alternate (for WAN Reroute). |
| Disable | Disables WRS, an individual secondary-circuit mapping, or alternate-circuit mapping. |
| Enable | Enables WRS, an individual secondary-circuit mapping, or alternate-circuit mapping. |
| List | Displays the current Restoral configuration. |
| Remove | Removes a primary to secondary mapping or a primary to alternate mapping created by add. |
| Set | Sets the values for the stabilization and time-of-day-revert-back timers. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Add

Use the **add** command to identify a secondary or an alternate dial-circuit or leased link interface for a primary serial link.

**Syntax:**

**add** alternate-circuit
secondary-circuit

**alternate-circuit**

The **add alternate-circuit** command binds an alternate interface to a primary interface for WAN Reroute purposes. You can assign multiple primaries to a single alternate interface. The alternate link type need not be

the same as the primary link type (for example, the alternate link type can be a PPP dial circuit and the primary link type can be a Frame Relay leased line).

**Example:**

```
WRS Config>add alt
Alternate interface number  [0]? 6
Primary interface number  [0]? 1
```

**Alternate interface number**
This is the interface number previously assigned to the alternate interface. Any LAN interface, PPP, Frame Relay, or X.25 serial interface, or a PPP or Frame Relay dial circuit is an eligible alternate 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 LAN interface, PPP, Frame Relay, or X.25 serial interface, or a PPP or Frame Relay dial circuit. The default is 0.

**secondary-circuit**
The **add secondary-circuit** command binds a secondary interface to a primary interface for WAN Restoral purposes. Both interfaces must have previously been configured. You can only assign one secondary interface to a primary and vice-versa.

**Example:**

```
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 PPP dial circuit or Multilink PPP interface can be a 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 a primary/secondary pairing for WAN Restoral, or to disable a primary/alternate pairing for WAN Reroute, or to disable Dial-on-overflow for a primary/alternate pairing.

**Syntax:**

**disable** alternate-circuit
dial-on-overflow
secondary-circuit
wrs

**alternate-circuit** *interface#*
Disables the primary/alternate pairing for WAN Reroute.

**Example:**

```
WRS Config> disable alternate-circuit
Alternate interface number  [0]? 6
```

**Alternate interface number**
This is the number of the alternate interface previously configured with the **add alternate-circuit** command. The default is 0.

**dial-on-overflow** *alt-intfc#*
Disables dial-on-overflow for all primary/alternate pairings using a specified alternate.

**Example:**

```
WRS Config> disable dial-on-overflow
alternate interface number  [0]? 6
```

**Alternate interface number**
This is the number of the alternate interface previously configured with the **add alternate-circuit** command. The default is 0.

**secondary-circuit** *interface#*
Disables the restoral of a particular primary interface by its associated secondary interface until the next **enable secondary-circuit** command at the WRS console. Both interfaces must have been previously configured and bound together in the WRS configuration.

**Example:**

```
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-circuit** command. The default is 0.

**wrs** Disables the WAN Restoral feature globally on the router. This means that WAN Reroute and Dial-on-overflow are also disabled.

## Enable

Use the **enable** command to enable the WAN Restoral function, to enable a primary/secondary pairing for WAN Restoral, to enabled a primary/alternate pairing for WAN Reroute, or to enable dial-on-overflow for a primary/alternate pairing.

**Syntax:**

**enable** alternate-circuit
dial-on-overflow
secondary-circuit
wrs

**alternate-circuit** *interface#*
Enables an alternate circuit

**Example:**

```
WRS Config>enable alternate-circuit
Alternate interface number  [0]? 6
```

**Alternate interface number**
This is the number of the alternate interface previously configured with the **add alternate-circuit** command. The default is 0.

**dial-on-overflow**

Enables dial-on-overflow and allows you to set parameters that control how dial-on-overflow works.

**Example:**

```
WRS>enable dial-on-overflow

For dial-on-overflow, only IP traffic can overflow to the alternate
interface.
Primary interface number  ]0]? 1
add-threshold (1-100% utilization) [90]?
drop-threshold(0-99% utilization) [60]?
bandwidth test interval(10-200 seconds) [15]?
minimum time to keep the alternate up (20-21600 sec.) [300]?
Dial-on overflow is enabled.
Remember to configure the primary interface's line speed!
```

**Primary interface number**

This is the interface number of the primary interface for which you are enabling dial-on-overflow. The default is 0.

**add-threshold**

Determines when an alternate interface will be brought up for additional bandwidth. This value must expressed as a percentage of the primary interface's configured line speed. The default is 90%.

**drop-threshold**

Determines when an alternate interface is no longer needed for additional bandwidth. This value must expressed as a percentage of the primary interface's configured line speed. The default is 60%.

**bandwidth monitoring interval**

Determines how often the primary interface's bandwidth is monitored for the *add-threshold* and *drop-threshold.* The default is 15 seconds.

**Minimum time to keep alternate up**

This time period needs to include enough time for the routers to establish the new route when IP traffic on the local router is rerouted to the alternate interface. The default is 5 minutes.

**secondary-circuit** *interface#*

Enables the restoral of a primary link by the indicated secondary link.

**Example:**

```
WRS Config>enable secondary-circuit
Secondary interface number  [0]? 3
```

**Secondary interface number**

This is the number of the secondary interface previously configured with the **add secondary-circuit** command. The default is 0.

**wrs** Enables the function of the WAN Restoral feature on the router. This means that if WAN Reroute and Dial-on-overflow are configured they are also enabled.

## List

Use the **list** command to display global configuration information for the feature and display configuration information for WAN Restoral primary-secondary pairs, WAN Reroute primary-alternate pairs, and Dial-on-Overflow.

**Syntax:**

**list**

**Example:**

```
WRS Config>list
WAN Restoral is enabled.
Default Stabilization Time:       0 seconds
Default First Stabilization Time: 0 seconds

                                                  Secondary
   Primary Interface       Secondary Interface    Enabled
----------------------  ----------------------  --------
 4 - WAN PPP             7 - PPP Dial Circuit       No
                                          Alt.     1st  Subseq TOD Revert Back
Primary Interface  Alternate Interface    Enabled  Stab  Stab  Start    Stop
-----------------  ---------------------  -------  ----  ----  -------  -------

 1 - WAN Frame Re   2 - WAN Frame Relay      Yes    dflt  dflt  Not Set  Not Set


Dial-on-overflow is enabled.
Primary    add-       drop-      test       minimum
Interface  threshold  threshold  interval   alt up time
---------  ---------  ---------  ---------  -----------
     1         29%        20%      15 sec.   300 sec.
```

## Remove

Use the **remove** command to delete the mapping of an alternate interface or secondary (backup) interface to the primary interface.

**Syntax:**

**remove** alternate-circuit
secondary-circuit

**alternate-circuit** *alternate-interface# primary-interface#*

Removes the mapping of a alternate (backup) interface to the primary interface for WAN Reroute. Both interfaces must have been previously assigned and bound together using the **add alternate-circuit** command.

**Alternate-interface#**

This is the number of the alternate interface previously configured with the **add alternate-circuit** command. The default is 0.

**Primary-interface#**

This is the interface number of the primary interface previously bound to the alternate being removed. The default is 0.

**Example:**

```
WRS Config> remove alternate-circuit
Alternate interface number  [0]? 3
Primary interface number  [0]? 1
```

**secondary-circuit** *secondary-interface# primary-interface#*

Removes the mapping of a secondary (backup) interface to the primary interface for WAN Restoral. Both interfaces must have been previously assigned and bound together using the **add secondary-circuit** command.

**Secondary-interface#**

This is the number of the secondary interface previously configured with the **add secondary-circuit** command. The default is 0.

**Primary-interface#**

This is the interface number of the primary interface previously bound to the secondary being removed. The default is 0.

**Example:**

```
WRS Config> remove secondary-circuit
Secondary interface number  [0]? 3
Primary interface number  [0]? 1
```

# Set

Use the **set** command to set the parameters for WAN Reroute.

**Syntax:**

**set ?** default

first-stabilization

stabilization

start-time-of-day-revert-back

stop-time-of-day-revert-back

**default**

Use the **set default** command to set the defaults to be used by links that do not have configured stabilization and first-stabilization times.

**first-stabilization**

Sets the default first-stabilization value to be used for links for which a first-stabilization time was not configured.

```
WRS Config>set default first
Default first primary stabilization time (0 - 3600 seconds)  [0]? 20
```

**stabilization**

Sets the default stabilization value to be used for links for which a stabilization time was not configured.

```
WRS Config>set default stab
 Default primary stabilization time (0 - 3600 seconds)  [0]? 30
```

**first-stabilization**

Sets the number of seconds at router initialization before routing for this primary link is switched to the alternate link if the primary link is not up.

**Example:**

```
WRS Config>set first
Primary interface number  [0]? 1
First primary stabilization time (0 - 3600 seconds -1 = default)  [-1]?
```

**Primary interface number**

This is the primary interface number of the primary interface for which you are setting first-stabilization. The default is 0.

**First primary stabilization time**

The stabilization time for this primary interface. The default is 1.

**stabilization**

Sets the number of seconds required after the primary link is first detected to be up before routing is switched back to the primary. Routing over the alternate link continues until the primary link remains up for this number of seconds.

**Example:**

```
WRS Config>set first
Primary interface number  [0]? 1
Primary stabilization time (0 - 3600 seconds -1 = default)  [-1]?
```

**Primary interface number**

This is the primary interface number of the primary interface for which you are setting stabilization. The default is 0.

**Primary stabilization time**
The stabilization time for the primary interface. The default is 1.

**start-time-of-day-revert-back**
The earliest time of the day the router can switch back to the primary route. The router can revert back to the primary any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary will only occur if the primary is up and the stabilization parameters are met. The default is 0.

**Example:**

```
WRS Config>set start
Primary interface number  [0]? 1
Time-of-Day revert back window start (1 - 24 hours, 0 = not configured) [0] 3
Start time-of-day revert back configured. Remember to configure stop time-of-day
```

**Primary interface number**
This is the primary interface number of the primary interface for which you are setting first-stabilization. The default is 0.

**Time-of-day-revert-back-window start**
This time marks the beginning time for the revert back window. The router can revert back to the primary interface any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary interface will only occur if the primary interface is up and the stabilization parameters are met. The default is 1.

**stop-time-of-day-revert-back**
This time marks the ending time for the revert back window. The router can revert back to the primary interface any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary interface will only occur if the primary interface is up and the stabilization parameters are met. The default is 1.

**Example:**

```
WRS Config>set stop
Primary interface number  [0]? 1
Time-of-Day revert back window stop (1 - 24 hours, 0 = not configured) [0]?5
```

**Primary interface number**
This is the primary interface number of the primary interface for which you are setting first-stabilization. The default is 0.

**Time-of-day-revert-back-window stop**
This time marks the ending time for the revert back window. The router can revert back to the primary interface any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary interface will only occur if the primary interface is up and the stabilization parameters are met. The default is 1.

## Accessing the WAN Restoral Interface Monitoring Process

To access the WAN Restoral interface monitoring process, enter the following command at the GWCON (+) prompt:

```
+ feature wrs
```

# WAN Restoral Monitoring Commands

The WAN Restoral (WRS) monitoring commands allow you to monitor the state of WAN Restoral primary-secondary pairs, WAN Reroute primary-alternate pairs, and Dial-on-Overflow. Any modifications to the operational state of WAN Restoral, WAN Reroute, and Dial-on-Overflow made through the monitoring interface are not maintained across router restarts.

Access the WRS prompt by entering **feature wrs** at the GWCON (+) prompt. Table 106 lists the WRS commands and their functions, and the following sections explain the commands.

*Table 106. WAN Restoral Monitoring Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Clear | Clears the monitoring statistics displayed using the **list** command. |
| Disable | Disables the WRS, or an individual secondary, or alternate, or dial-on-overflow. |
| Enable | Enables the WRS, or an individual secondary, or alternate, or dial-on-overflow. |
| List | Displays the monitoring information on one or all alternate or secondary circuits. |
| Set | Sets the values for the stabilization and time-of-day-revert-back-timers. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Clear

Use the **clear** command to clear WAN Restoral, WAN Reroute, and dial-on-overflow statistics that are displayed using the **list** command.

**Syntax:**

**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, disable the restoral of a particular primary interface by its associated secondary interface, disable an alternate interface or disable dial-on-overflow.

**Syntax:**

**disable** alternate-circuit
dial-on-overflow
secondary-circuit
wrs

**alternate-circuit**
Disables a primary/alternate pairing for WAN Reroute. There can be multiple pairings using the same alternate. This command disables all the pairings using the specified alternate-circuit.

**Example:**

```
WRS>disable alternate-circuit
Alternate circuit number  [0]? 6
```

**Alternate circuit number**
This is the number of the alternate circuit. The default is 0.

**dial-on-overflow**
Disables dial-on-overflow for the specified primary/alternate pairing, without changing the enabled/disabled state of WAN Reroute for that pairing. If dial-on-overflow is actively routing, it is terminated at the expiration of the next monitor interval.

**secondary-circuit**
Disables the restoral of a particular primary interface by its associated secondary interface until the next **restart**, **reload**, or **enable secondary-circuit** command. Both interfaces must have been previously configured and bound together in the WRS configuration.

Normally, in **talk 5** (GWCON), 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-circuit** at the WAN Restoral monitoring prompt and disable the secondary interface at the top level GWCON prompt.**Example:**

```
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-circuit** command. The default is 0.

**wrs** Disabling WRS disables WAN Restoral, WAN Reroute, and Dial-on-overflow on the router until the next **restart**, **reload**, or **enable WRS** command.

## Enable

Use the **enable** command to enable the WAN Restoral interface, enable the restoral of a primary link by a secondary circuit, enable an alternate circuit, or enable dial-on-overflow.

**Syntax:**

**enable** alternate-circuit
dial-on-overflow
secondary-circuit
wrs

**alternate-circuit**
Enables the primary/alternate pairings for WAN Reroute for all pairings using the specified alternate.

Example:

```
WRS> enable alternate-circuit
Alternate circuit number  [0]? 3
```

**Alternate circuit number**
This is the interface number of the alternate circuit. The default is 0.

**dial-on-overflow**
Enables dial-on-overflow and allows you to set parameters that control dial-on-overflow. Optionally, allows you to cause the IP protocol to be switched immediately to the alternate, as if the add threshold had been crossed.

**Example:**

```
WRS> dial-on-overflow

For dial-on-overflow, only IP traffic can overflow to the alternate interface.
Primary interface number  [0]? 1
add-threshold (1-100% utilization) [90]?
drop-threshold(0-99% utilization) [60]?
bandwidth test interval(10-200 seconds) [15]?
minimum time to keep the alternate up (20-21600 sec.) [300]?
Dial-on overflow is enabled.
Remember to configure the primary interface's line speed!

Do you want to switch IP traffic to the alternate now?(Yes or [No]):
WRS>
```

**secondary-circuit**
Enables the restoral of a primary link by the indicated secondary link.

**Example:**

```
WRS> enable secondary-circuit
Secondary interface number  [0]? 3
```

**Secondary interface number**
This is the number of the secondary interface previously configured with the **add secondary-circuit** command. The default is 0.

**wrs** Enables the function of the WAN Restoral feature on the router. This feature needs to be enabled in order to do WAN Restoral, WAN Reroute, or Dial-on-overflow.

## Set

Use the **set** command to set the parameters for WAN Reroute.

**Syntax:**

**set ?**

default

first-stabilization

stabilization

start-time-of-day-revert-back

stop-time-of-day-revert-back

**default**
Use the **set default** command to set the defaults to be used by links that don't have configured stabilization and first-stabilization times.

**Example:**

```
WRS Config>set default ?
FIRST-STABILIZATION
STABILIZATION
```

**first-stabilization**

Sets the default first-stabilization value to be used for links for which a first-stabilization time was not configured.

```
WRS Config>set default first

Default first primary stabilization time (0 - 3600 seconds)  [0]? 20
```

**stabilization**

Sets the default stabilization value to be used for links for which a stabilization time was not configured.

```
WRS Config>set default stab
Default primary stabilization time (0 - 3600 seconds)  [0]? 30
```

**first-stabilization**

Sets the number of seconds at router initialization before routing for this primary link is switched to the alternate link if the primary link is not up.

**Example:**

```
WRS Config>set first
Primary interface number  [0]? 1
First primary stabilization time (0 - 3600 seconds -1 = default)  [-1]?
```

**Primary interface number**

This is the primary interface number of the primary interface for which you are setting first-stabilization. The default is 0.

**First primary stabilization time**

The stabilization time for this primary interface. The default is 1.

**stabilization**

Sets the number of seconds required after the primary link is first detected to be up before routing is switched back to the primary. Routing over the alternate link continues until the primary link remains up for this number of seconds.

**Example:**

```
WRS Config>set first
Primary interface number  [0]? 1
Primary stabilization time (0 - 3600 seconds -1 = default)  [-1]?
```

**Primary interface number**

This is the primary interface number of the primary interface for which you are setting stabilization. The default is 0.

**Primary stabilization time**

The stabilization time for the primary interface. The default is 1.

**start-time-of-day-revert-back**

The earliest time of the day the router can switch back to the primary route. The router can revert back to the primary any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary will only occur if the primary is up and the stabilization parameters are met. The default is 0.

**Example:**

```
WRS Config>set start
Primary interface number [0]? 1
Time-of-Day revert back window start (1 - 24 hours, 0 = not configured) [0] 3
Start time-of-day revert back configured. Remember to configure stop time-of-day
```

**Primary interface number**

This is the primary interface number of the primary interface for which you are setting first-stabilization. The default is 0.

**Time-of-day-revert-back-window start**

This time marks the beginning time for the revert back window. The

router can revert back to the primary interface any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary interface will only occur if the primary interface is up and the stabilization parameters are met. The default is 1.

**stop-time-of-day-revert-back**

This time marks the ending time for the revert back window. The router can revert back to the primary interface any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary interface will only occur if the primary interface is up and the stabilization parameters are met. The default is 1.

**Example:**

```
WRS Config>set stop
Primary interface number  [0]? 1
Time-of-Day revert back window start (1 - 24 hours, 0 = not configured) [0]?
5
```

**Primary interface number**

This is the primary interface number of the primary interface for which you are setting first-stabilization. The default is 0.

**Time-of-day-revert-back-window stop**

This time marks the ending time for the revert back window. The router can revert back to the primary interface any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary interface will only occur if the primary interface is up and the stabilization parameters are met. The default is 1.

## List

Use the **list** command to display monitoring information on one or all WAN Restoral primary-secondary pairs or one or all WAN Reroute primary-alternate pairs.

**Syntax:**

**list** all | alternate-circuit | secondary-circuit | summary

**all** Provides summary information, followed by the specific information, for each secondary interface.

**Example:**

```
list all
WAN Restoral/Re-route is enabled with 2 circuits configured
    Total restoral attempts =          7 completions   =          7
    Total packets forwarded =         39
    Longest completed restoral period in hrs:min:sec     0:03:27

    Total overflow attempts =         20 completions   =         19
    Longest completed overflow period in hrs:min:sec     0:05:00

   Primary       Secondary    Restoral  Restoral   Current/Longest
Net Interface  Net Interface  Enabled    Active       Duration
-------------  -------------  --------  --------   --------------------
  4  PPP/0       7  PPP/1       No         No       00:03:27/ 00.06.00

                              Re-route/ Re-route/         Recent
   Primary       Alternate    Overflow  Overflow     Reroute/Overflow
```

```
Net Interface  Net Interface  Enabled    Active          Duration
-------------  -------------  --------  --------   --------------------
  1   FR/0       2   FR/1     Yes/Yes    No /No     00:00:56/ 00:05:00
```

**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 Completed 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.

**Total Overflow Attempts**
The number of attempts due to an overflow.

**Completions**
The number of successful overflow attempts when the secondary link came up and was used.

**Longest Completed Overflow Period**
Displays in hours, minutes , and seconds the longest amount of time an overflow was in operation, not counting any current usage.

**Primary Net Interface**
The interface that is being backed up by its associated secondary interface.

**Secondary Net Interface**
The dial circuit that is being used to back up the associated primary interface.

**Restoral Enabled**
Indicates that restoral of this primary interface is currently enabled.

**Restoral Active**
Indicates whether restoral is active (Yes or No).

**Current/Longest Duration**
Indicates in hours, minutes, and seconds the current and longest duration the secondary net interface was up.

**Primary Net Interface**
The interface that is being backed up by its associated alternate interface.

**Alternate Net Interface**
The interface that is being used as an alternate back up the associated primary interface.

**Re-route/Overflow Enabled**
Indicates whether reroute and overflow are enabled (Yes or No).

**Re-route/Overflow Active**
Indicates whether reroute and overflow are active (Yes or No).

**Recent Re-route Overflow Duration**
Indicates in hours, minutes, and seconds the recent reroute and overflow duration of the alternate net interface.

**Alternate-circuit**
Provides totals for an alternate circuit. Allows the monitoring operator to retrieve the WAN Reroute state and associated statistics for each alternate interface and its associated primary mapping.

**Example:**

```
WRS>li alt 7
Primary    1:FR/0 Frame Relay V.35/V.36
Alternate  7:PPP/1 Point to Point V.25bis Dial Circuit
reroute Enabled, currently inactive
overflow Enabled, currently inactive
Primary first stabilization time: default (0 seconds)
Primary stabilization time: default (0 seconds)
Time-of-day revert back not configured: start = 0, stop = 0
Restored 0 times (0 attempts)
Overflow 0 times (0 attempts)
```

**Primary Interface**
The interface that is being backed up by this associated alternate interface.

**Alternate Interface**
The dial circuit that is being used to back up the associated primary interface.

**Reroute Enabled**
Indicates whether reroute of this primary interface is currently enabled.

**Overflow Enabled**
Indicates whether overflow of this primary interface is currently enabled.

**Primary first stabilization**
The number of seconds at router initialization before routing for this primary link is switched to the alternate link if the primary link is not up.

**First stabilization**
The number of seconds required after the primary link is first detected to be up before routing is switched back to the primary. Routing over the alternate link continues until the primary link remains up for this number of seconds.

**Time-of-day revert back**
The time of the day the router can switch back to the primary route. The router can revert back to the primary any time between the start-time-of-day-revert-back time and the stop-time-of-day-revert-back time. Reverting back to the primary will only occur if the primary is up and the stabilization parameters are met. The default is 0.

**Restored times**
The number of attempts to reroute the primary interface.

**Overflow times**
The number of dial-on-overflow attempts.

**secondary-circuit**
Provides totals for each secondary circuit. Allows the monitoring 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.

# Chapter 63. The WAN Reroute Feature

This chapter describes the WAN reroute feature. It includes the following sections:



## WAN Reroute Overview

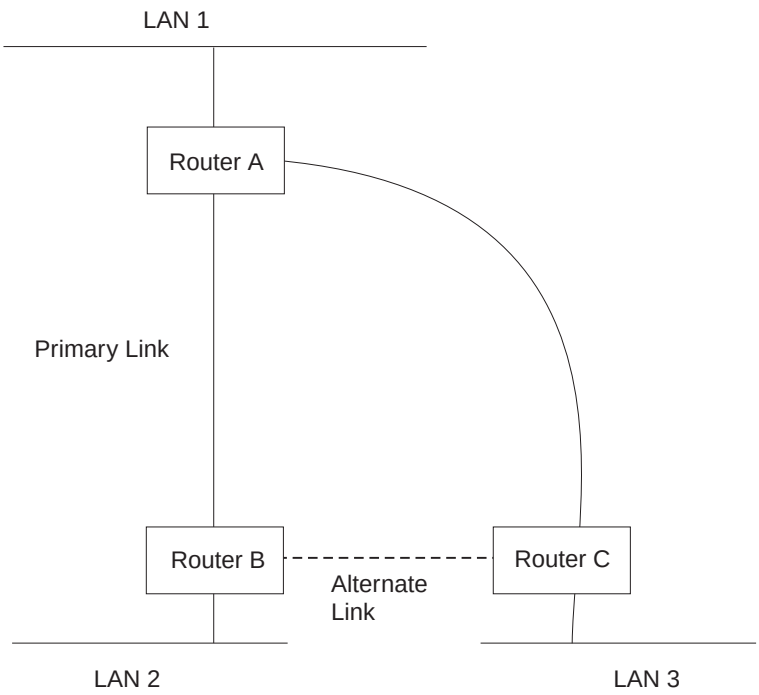
WAN Reroute lets you set up an alternate route so that if a primary link fails, the router automatically initiates a new connection to the destination through the alternate route. See "Overview for WAN Restoral, WAN Reroute, and Dial-on-Overflow" on page 739 for an explanation of WAN Restoral, and how WAN Reroute and Dial-on-overflow work together.

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

The alternate link can be any link on which you can configure routable protocols (for example, IP, IPX) and the datalink type of the alternate link need not match the datalink type of the primary link. For example, the alternate link can be a LAN interface, a PPP, Frame Relay, or X.25 serial interface, or a PPP or Frame Relay dial circuit. The following are examples of interface types that cannot be alternate links: SDLC serial interfaces, SRLY serial interfaces, and base nets like V.25bis and ISDN.

**Note:** If the primary link or alternate link is a dial circuit, that dial circuit cannot be configured for dial-on-demand.




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.

*Figure 52. WAN Reroute. Normally, there is a connection between Routers A and B and Routers A and C.*

# Dial-on-Overflow

Dial-on-overflow allows you to use an alternate interface for IP traffic when the traffic rate on the primary link reaches a specified threshold. This means that the primary interface does not have to be down before the alternate link is brought up. When the primary interface's traffic reaches the specified threshold the router brings up the alternate link. To use dial-on-overflow, WAN Reroute must be configured and the primary interface must be Frame Relay. IP is the only protocol that can be switched over to the alternate interface by dial-on-overflow. Also, OSPF should be used as the IP routing protocol instead of RIP when dial-on-overflow is used.

For information about configuring dial-on-overflow, see "WAN Restoral, WAN Reroute, and Dial-on-Overflow Configuration Commands" on page 743.

## Bandwidth Monitoring

The interval for bandwidth monitoring can be specified for dial-on-overflow during WAN Reroute configuration. The primary interface's receive and transmit bandwidth utilization are monitored. When the primary interface's bandwidth reaches the *add* threshold, a WAN Reroute request is generated to bring up the alternate interface. If WAN Reroute is successful bringing up the alternate interface, IP stops routing over the primary interface and starts routing over the alternate interface.

If WAN Reroute is not successful in bringing up the alternate route it periodically attempts to bring up the alternate interface until the primary interface's bandwidth utilization drops below the *drop* threshold.

When the primary interface's receive and transmit bandwidth utilization reaches the *drop* threshold and the minimum configured up time has expired the alternate interface is dropped. This causes IP to stop routing over the alternate interface and start using the primary interface.

The add-threshold and the drop-threshold are specified as a percentage of the configured line speed for the primary link. The configured line speed does not always match the actual speed of the link. The amount of traffic on the link in each direction is calculated separately. The threshold is exceeded if the traffic in either direction is greater then the specified percentage.

## 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 datalink types. The primary and alternate links can be:
   - A LAN interface.
   - A PPP serial interface.
   - A Frame Relay serial interface.
   - An X.25 serial interface.
   - A PPP dial circuit.
   - A Frame Relay dial circuit.

## Sample WAN Reroute Configuration

Figure 53 on page 762 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. If one of the primary links from a branch to headquarters fails, WAN reroute establishes an alternate route to headquarters through another branch.

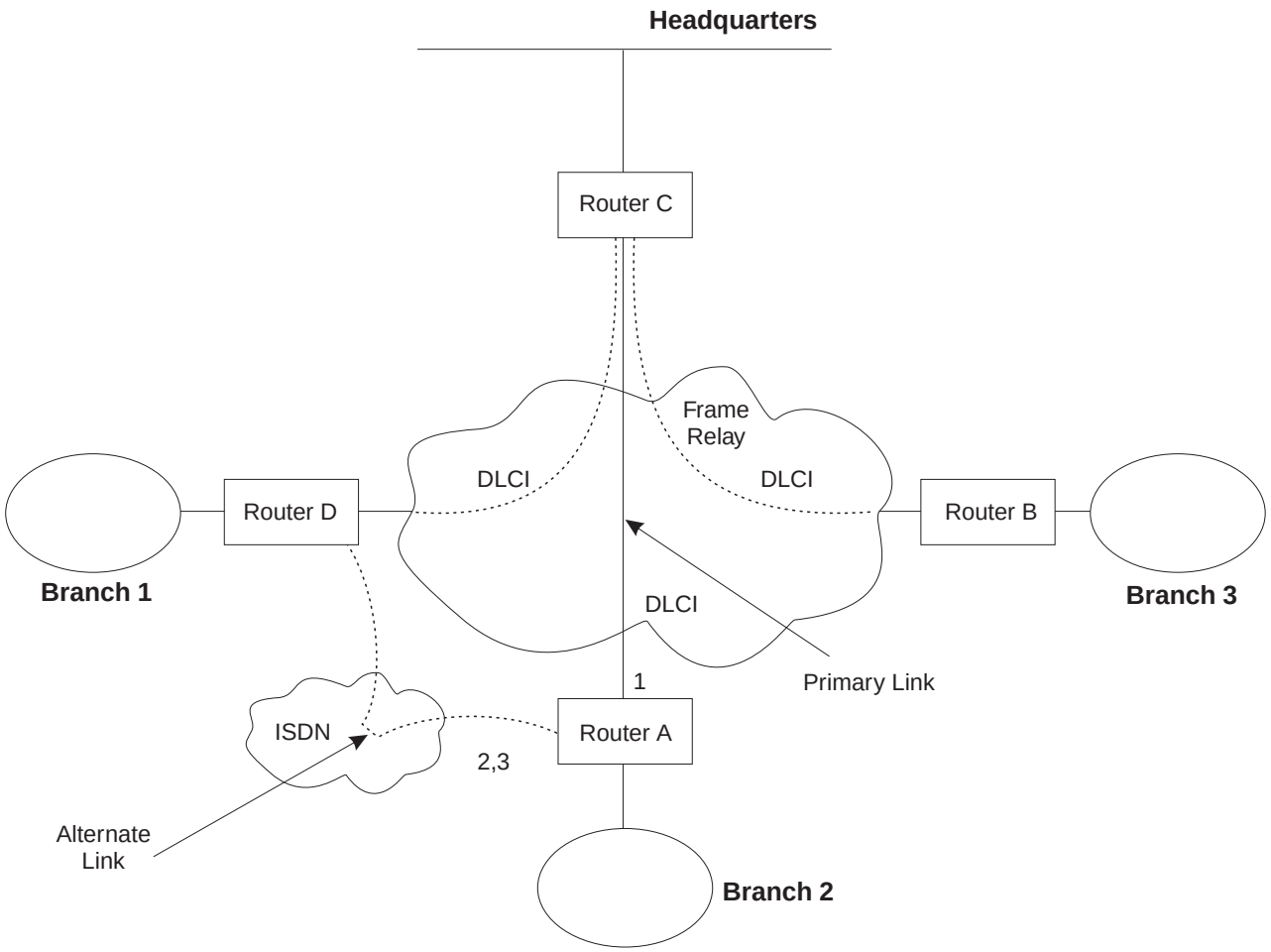



*Figure 53. Sample WAN Reroute Configuration. Branch offices use frame relay to connect to headquarters.*

The following sections describe how to set up WAN reroute on Router A in Figure 53. 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 and issue the 'set idle 0' command at the dial circuit config prompt.
  - 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 39. Using Frame Relay Interfaces" on page 457 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 39. Using Frame Relay Interfaces" on page 457.

## Configuring the ISDN Interface and Dial Circuit

Configure the ISDN interface and dial circuit between Router A and Router D. See "Chapter 51. Using the ISDN Interface" on page 629 for information on how to configure ISDN interfaces and dial circuits.

Unlike WAN Restoral, you must configure routable protocols on the dial circuit that will be used as the alternate link. If those routable 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.

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 a LAN interface, a PPP, Frame Relay, or X.25 serial interface, or a PPP or Frame Relay dial circuit, 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 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 1 a.m. 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 11 p.m. until 3 a.m.

   b. A start time of 1 and a stop time of 5 will give a revert-back window from 1 a.m. to 5 a.m.

```
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]?
```

## Configuring WAN Reroute

# Chapter 64. Using the Network Dispatcher Feature

This chapter describes how to use the Network Dispatcher Feature and contains the following sections:



For additional information about Network Dispatcher, see *Interactive Network Dispatcher User's Guide, GC31-8496*.

## Overview of Network Dispatcher

Network Dispatcher is a feature that boosts the performance of servers by forwarding TCP/IP session requests to different servers within a group of servers, thus load balancing the requests among all servers. The forwarding is transparent to the users and other applications. Network Dispatcher is useful for applications such as e-mail, servers, World Wide Web servers, distributed parallel database queries, and other TCP/IP applications.

Network Dispatcher can help maximize the potential of your site by providing a powerful, flexible, and scalable solution to peak-demand problems. During peak demand periods, Network Dispatcher can automatically find the optimal server to handle incoming requests.

The Network Dispatcher function does not use a domain name server for load balancing. It balances traffic among your servers through a unique combination of load balancing and management software. Network Dispatcher can also detect a failed server and forward traffic to other available servers.

All client requests sent to the Network Dispatcher machine are forwarded to the server that is selected by the Network Dispatcher as the optimal server according to certain dynamically set weights. You can use the default values for those weights or change the values during the configuration process.

The server sends a response back to the client without any involvement of Network Dispatcher. No additional software is required on your servers to communicate with Network Dispatcher.

The Network Dispatcher function is the key to stable, efficient management of a large, scalable network of servers. With Network Dispatcher, you can link many individual servers into what appears to be a single, virtual server. Your site thus appears as a single IP address to the world. Network Dispatcher functions independently of a domain name server; all requests are sent to the IP address of the Network Dispatcher machine.

Network Dispatcher brings distinct advantages in load balancing traffic to clustered servers, resulting in stable and efficient management of your site.

## Balancing TCP/IP Traffic Using Network Dispatcher

There are many different approaches to load balancing. Some of these approaches allow users to choose a different server at random if the first server is slow or not responding. Another approach is round-robin, in which the domain name server selects a server to handle requests. This approach is better, but does not take into consideration the current load on the target server or even whether the target server is available.

Network Dispatcher can load balance requests to different servers based on the type of request, an analysis of the load on servers, or a configurable set of weights that you assign. To manage each different type of balancing, the Network Dispatcher has the following components:

**Executor**
Load balances connections based on the type of request received. Typical requests types are HTTP, FTP, and SSL. This component always runs.

**Advisors**
Queries the servers and analyzes the results by protocol for each server. The advisor passes this information to the ***manager*** to set the appropriate weight. The advisor is an optional component.

Network Dispatcher supports advisors for FTP and HTTP as well as an MVS advisor that works with Workload Manager (WLM) on MVS systems. WLM manages the amount of workload on an individual MVS ID. Network Dispatcher can use WLM to help load balance requests to MVS servers.

**Manager**
Sets weights for a server based on:

- Internal counters in the executor
- Feedback from the servers provided by the advisors
- Feedback from any system monitoring program

The manager is an optional component. However, if you do not use the manager, the Network Dispatcher will balance the load using a round-robin scheduling method based on the current server weights.

## High Availability for Network Dispatcher

The base Network Dispatcher function has the following characteristics that makes it a single point of failure from many different perspectives:

- It examines all the traffic on the way in. If some of the packets for an existing connection use a different path through a different Network Dispatcher to reach a server, the server immediately resets the connection.
- It keeps track of all established connections and although it does not terminate them, entries lost from the Network Dispatcher connection table will result in the resetting of a connection.
- It appears to any previous hop router as the last hop, and the connection's termination.

All these characteristics make the following failures critical for the whole cluster:

- If the Network Dispatcher fails for any reason, all the connection tables are lost, therefore all existing connections from the client to the server are also lost.

Assuming there is a second Network Dispatcher that can direct a client to the servers, new connections will be able to go through only after the usual routing protocol delays which could be several minutes.

- If the configured Network Dispatcher interface to the previous IP router fails, there must either be another interface to get to the same Network Dispatcher, in which case recovery is performed by the IP router (using the ARP aging mechanism with delays in the order of several minutes), or all connections will be lost.
- If Network Dispatcher interface to the servers fails, the previous hop router assumes that the Network Dispatcher is the last hop, and therefore will not reroute new connections. Existing connections will be lost and new connections will not be established.

In all these failure cases, which are not only Network Dispatcher failures but also Network Dispatcher neighborhood failures, all the existing connections are lost. Even with a backup Network Dispatcher running standard IP recovery mechanisms, recovery is, at best, slow and applies only to new connections. In the worst case, there is no recovery of the connections.

To improve Network Dispatcher availability, the Network Dispatcher High Availability function uses the following mechanisms:

- Two Network Dispatchers with connectivity to the same clients, and the same cluster of servers, as well as connectivity between the Network Dispatchers.
- A “Heartbeat” mechanism between the two Network Dispatchers to detect Network Dispatcher failure.
- A reachability criteria, to identify which IP host can and cannot be reached from each Network Dispatcher.
- Synchronization of the Network Dispatcher databases (that is, the connection tables, reachability tables, and other databases).
- Logic to elect the active Network Dispatcher, which is in charge of a given cluster of servers, and the standby Network Dispatcher, which continuously gets synchronized for that cluster of servers.
- A mechanism to perform fast IP takeover, when the logic or an operator decides to switch active and standby.

## Failure Detection

Besides the basic criteria of failure detection, (the loss of connectivity between active and standby Network Dispatchers, detected through the Heartbeat messages) there is another failure detection mechanism named “reachability criteria.” When you configure the Network Dispatcher, you provide a list of hosts that each of the Network Dispatchers should be able to reach to work correctly. The hosts could be routers, IP servers or other types of hosts. Host reachability is obtained by pinging the host.

Switchover takes place either if the Heartbeat messages cannot go through, or if the reachability criteria are no longer met by the active Network Dispatcher and the standby Network Dispatcher is reachable. To make the decision based on all available information, the active Network Dispatcher regularly sends the standby Network Dispatcher its reachability capabilities. The standby Network Dispatcher then compares the capabilities with its own and decides whether to switch.

## Cache Synchronization

The main data synchronized by the Network Dispatchers are the connection table entries. The Network Dispatcher High Availability function uses a cache synchronization protocol that insures that both Network Dispatchers contain the same entries. This synchronization takes into account a known error margin of transmission delays. The protocol performs an initial synchronization of peer databases and later, maintains the databases through periodic updates.

## Recovery Strategy

In the case of a Network Dispatcher failure, the IP takeover mechanism will promptly direct all traffic toward the standby Network Dispatcher. The Database Synchronization mechanism insures that the standby has the same entries as the active Network Dispatcher. When the failure occurs in the network (any intermediate piece of hardware or software between the client and the back-end server), and there is an alternate path through the standby Network Dispatcher that works, the switchover is performed across the alternate path.

## IP Takeover

**Note:** Cluster IP Addresses are assumed to be on the same logical subnet as the previous hop router (IP router).

The IP Router will resolve the cluster address through the ARP protocol. To perform the IP takeover, the Network Dispatcher (standby becoming active) will issue an ARP request to itself, that is broadcasted to all directly attached networks belonging to the logical subnet of the cluster. The previous hops' IP router will update their ARP tables (according to RFC826) to send all traffic for that cluster to the new active (previously standby) Network Dispatcher.

# Configuring Network Dispatcher

There are many ways that you can configure Network Dispatcher to support your site. If you have only one host name for your site to which all of your customers will connect, you can define a single cluster and any ports to which you want to receive connections. This configuration is shown in Figure 54 on page 771.

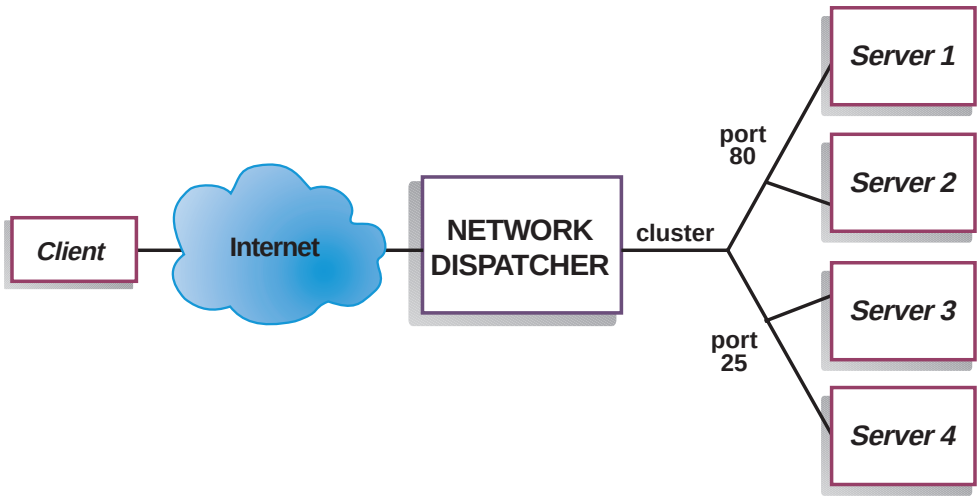



*Figure 54. Example of Network Dispatcher Configured With a Single Cluster and 2 Ports*

Another way of configuring Network Dispatcher would be necessary if your site does content hosting for several companies or departments, each one coming into your site with a different URL. In this case, you might want to define a cluster for each company or department and any ports to which you want to receive connections at that URL as shown in Figure 55 on page 772.

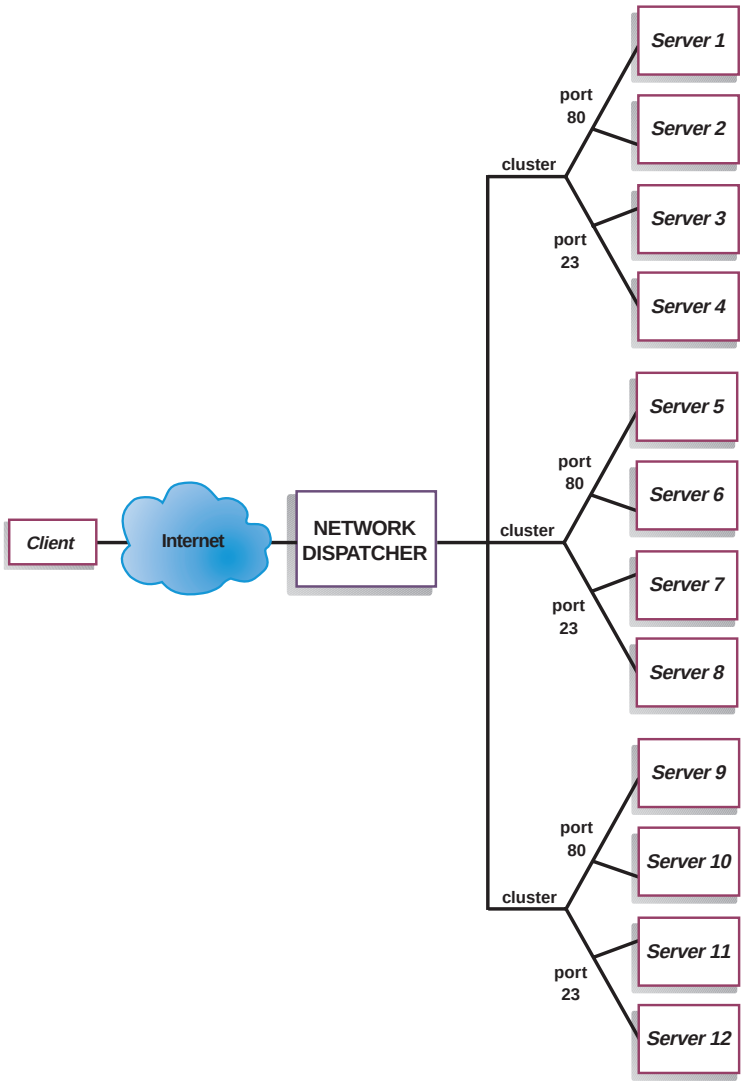



*Figure 55. Example of Network Dispatcher Configured With 3 Clusters and 3 URLs*

A third way of configuring Network Dispatcher would be appropriate if you have a very large site with many servers dedicated to each protocol supported. For example, you may choose to have separate FTP servers with direct T3 lines for large downloadable files. In this case, you might want to define a cluster for each protocol with a single port but many servers as shown in Figure 56 on page 773.

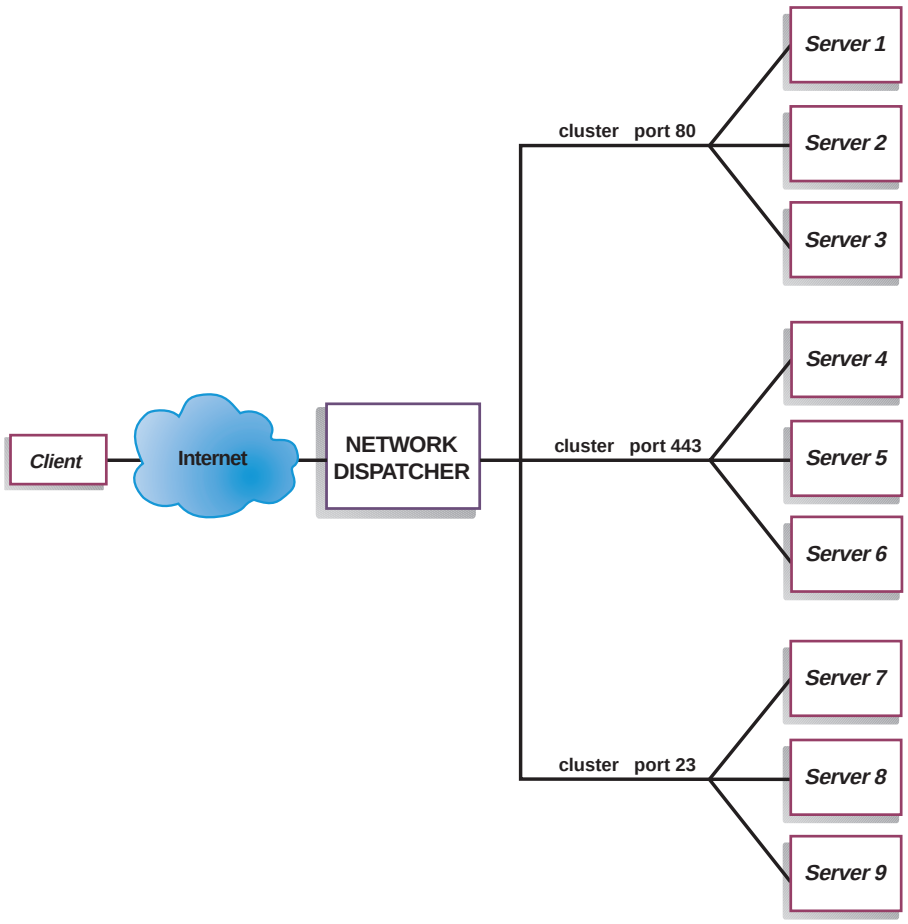



*Figure 56. Example of Network Dispatcher Configured with 3 Clusters and 3 Ports*

## Configuration Steps

Before configuring Network Dispatcher:

1. Make sure that the Network Dispatcher has direct interfaces to servers. Servers can have independent connections to the enterprise router or Internet, such that the outgoing traffic from servers to clients can bypass the Network Dispatcher; however, you do not have to configure the independent connection.

   If high availability is important for your network, a typical high availability configuration is shown in Figure 57 on page 774.

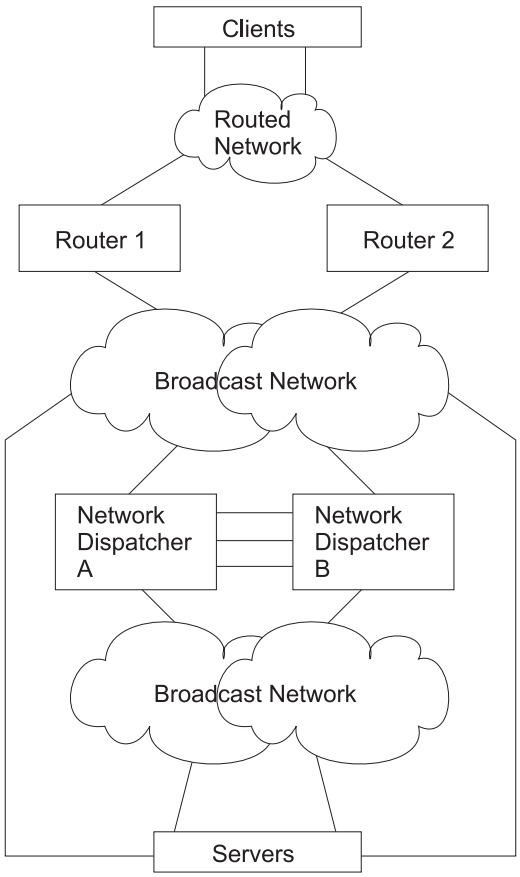



*Figure 57. High Availability Network Dispatcher Configuration*

2. Configure the interfaces of the device. This includes configuring all interfaces, IP addresses on all interfaces, and any applicable routing protocols. You must also configure an internal IP address, using the **set internal-ip-address** command. This is required if you plan to use the Manager and Advisors components. See *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1* for more information about the **set internal-ip-address.**
3. Reboot or restart the device.

## Configuring Network Dispatcher on a IBM 2216

To configure Network Dispatcher on a IBM 2216:

1. Access the Network Dispatcher feature, using the **feature ndr** command.
2. Enable the executor and the manager using the **enable executor** and **enable manager** commands.
3. Configure the clusters using the **add cluster** command.
4. Configure the TCP destination ports using the **add port** for each cluster of servers that will serve the corresponding protocol. Examples of the ports are: 80 for HTTP, 20 and 21 for FTP, and 23 for telnet.
5. Configure the servers using the **add server** commands. A server is always associated with a port and a cluster. A server can serve more than one port, a port can be served on more than one server, and a server can belong to more than one cluster, if the server's operating system supports multiple aliasing.
6. Configure any advisors using the **add advisor** command.

**Note:** For the MVS advisor, do not define port 10007 under any cluster. The advisor will search the list of all configured servers to find advisable servers.

7. Enable the advisors that you configured using the **enable advisor** command.

If you are configuring the Network Dispatcher for high availability, continue with the following steps. Otherwise, you have completed the configuration.

**Note:** Perform these steps on the primary Network Dispatcher and then on the backup.

8. Configure whether this Network Dispatcher is a primary or backup and whether the switchover is manual or automatic using the **add backup** command.
9. Configure all paths (more than one is recommended) on which the heartbeat is going to take place between the primary and backup Network Dispatchers using the **add heartbeat** command. A path is specified by source and destination IP addresses.
10. Configure the list of host IP addresses that the Network Dispatcher must be able to reach in order to insure a full service, using the **add reach** command. Typically, this will be a subset of servers, the enterprise router, or an administration station.

You can change the configuration using the **set**, **remove**, and **disable** commands.

## Configuring a Server for Network Dispatcher

To configure the Network Dispatcher on a server:

1. Alias the loopback device.

   For the TCP servers to work, you must set (or preferably alias) the loopback device (usually called ***lo0***) to the cluster address. Network Dispatcher does not change the destination IP address in the TCP/IP packet before forwarding the packet to a TCP server machine. When you set or alias the loopback device to the cluster address, the TCP server machine will accept a packet that was addressed to another machine.

   If you have an operating system that supports network interface aliasing such as AIX, Solaris, or Windows NT, you should alias ***lo0*** to the cluster address. The benefit of using an operating system that supports aliases is that you can configure the TCP server machines to serve multiple cluster addresses.

   If you have a server with an operating system that does not support aliases, such as HP-UX and OS/2, you must set ***lo0*** to the cluster address.

   If your server is an MVS system running TCP/IP V3R2, you must set the VIPA address to the cluster address. This will function as a loopback address. The VIPA address must not belong to a subnet that is directly connected to the MVS node. If your MVS system is running TCP/IP V3R3, you must set the loopback device to the cluster address.

2. Check for an extra route.

   The network mask for the loopback device is usually 255.0.0.0, so a default route will probably be created. This route needs to be removed.

   Check for an extra route on Windows NT with the **route print** command.

   Check for an extra route on all UNIX systems and OS/2 with the **netstat -nr** command.

3. Delete any extra routes.

   Use the command from Table 107 on page 776 for your operating system to delete any extra routes.

*Table 107. Commands to Delete Routes for Various Operating Systems*

| **Operating System** | **Command** |
|---|---|
| AIX | **route delete -net** *network_address cluster_address* |
| HP-Unix | **route delete net** *cluster_address* |
| Solaris | No need to delete route. |
| OS/2 | No need to delete route. |
| Windows NT | **route delete** *network_address cluster_address*<br>**Note:** This command should be entered at an MS-DOS prompt. |

# Chapter 65. Configuring and Monitoring the Network Dispatcher Feature

This chapter describes the Network Dispatcher Feature configuration and operational commands. It contains the following sections:



## Accessing the Network Dispatcher Configuration Commands

To access the Network Dispatcher configuration environment:

1. Enter **talk 6** at the OPCON prompt (*).
2. Enter **feature ndr** at the `Config >` prompt.

## Network Dispatcher Configuration Commands

Table 108 summarizes the Network Dispatcher configuration commands and the rest of the section explains these commands. Enter these commands at the `NDR Config >` prompt.

*Table 108. Network Dispatcher Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Add | Configures various components of the Network Dispatcher including advisors, clusters, ports, and servers. |
| Clear | Clears the entire Network Dispatcher configuration. |
| Disable | Disables the backup, executor, and manager components of the Network Dispatcher. Also disables specific advisors. |
| Enable | Enables the backup, executor, and manager components of the Network Dispatcher. Also enables specific advisors. |
| List | Displays the entire Network Dispatcher Configuration or specific portions of the configuration. |
| Remove | Removes specific portions of the Network Dispatcher configuration. |
| Set | Changes the configuration parameters for advisors, clusters, ports, servers, or the Network Dispatcher manager. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Add

Use the **add** command to configure advisors, clusters, ports, servers, and to specify which hosts or subnets are reachable through the Network Dispatcher. For High Availability you can also configure whether this Network Dispatcher is a primary or backup and which IP addresses to use for heartbeat and cache synchronization.

**Syntax:**

**add** advisor . . .

backup . . .

cluster . . .

heartbeat . . .

port . . .

reach . . .

server . . .

**Advisor** *name port interval timeout*

Specifies the name and port for an advisor. This parameter also specifies how frequently the advisor will collect information on a particular protocol and a time period after which the advisor considers the protocol unavailable.

**name** Specifies the type of advisor.

**Valid values:** 0, 1, 2

0 = FTP

1 = HTTP

2 = MVS

**Default value:** 1

**port** Specifies the port number for this advisor.

**Valid values:** 0 to 65535

**Default values:**

| Advisor Number | Default Value |
|---|---|
| **0** | 21 |
| **1** | 80 |
| **2** | 10007 |

**interval**

Specifies the frequency, in seconds, with which the advisor queries its protocol for each server. After half of this value without a response from the server, the adviser considers the protocol unavailable.

**Valid values:** 0 to 65535

**Default value:** 5

**timeout**

Specifies the interval of time, in seconds, after which the advisor considers the protocol unavailable.

To make sure that out-of-date information is not used by the manager in its load-balancing decisions, the manager will not use information from the advisor whose time stamp is older than the time set in this parameter. The advisor timeout should be larger than the advisor polling interval. If the timeout is smaller, the manager will ignore reports that should be used. By default, advisor reports do not time out.

This timeout value typically applies if you disable an advisor. Do not confuse this parameter with the interval/2 timeout previously described, which relates to a server not responding.

**Valid values:** 0 to 65535

**Default value:** 0, which means the protocol is considered always available.

**Example:**

```
add advisor
Advisor name (0=ftp, 1=http, 2=mvs) [1]? 1
Port number [80]?
Interval (seconds) [5]? 10
Timeout (0=unlimited) [0]? 10
```

**backup** *role strategy*

Specifies whether this Network Dispatcher is a backup or primary.

**role** Defines whether this is a primary or a backup Network Dispatcher. Use this command only if you intend to have a redundant configuration, and want the High Availability function to run. In this case, you must also configure the heartbeat (**add heartbeat**) and reachability (**add reach**).

**Valid values:** 0 or 1

0 = primary

1 = backup

**Default value:** 0

**strategy**

Specifies whether the Network Dispatcher will switch back to primary mode automatically or manually. Whenever a Primary Network Dispatcher fails and become standby (which means a backup performed the IP takeover function), and then becomes available, it will automatically become the active Network Dispatcher if the strategy is set to *automatic*, as soon as the caches are synchronized. If strategy is set to *manual*, the old primary will go to standby mode and the operator must use the **switchover** command to make it active again. See “Switchover” on page 799.

**Valid values:** 0 or 1

0 = automatic

1 = manual

**Default value:** 0

**Example:**

```
add backup
Role (0=Primary, 1=Backup) [0]?
Switch back strategy (0=Auto, 1=Manual) [0]?
```

**cluster** *address FIN-count FIN-timeout FIN-stale-timer*

Specifies a cluster’s IP address and the frequency for the executor to perform garbage collection from the Network Dispatcher database.

**address**

Specifies the IP address for the cluster.

**Valid values:** Any valid IP address

**Default value:** 0.0.0.0

**FIN-count**

Specifies the number of connections that must be in FIN state

before the executor tries to remove the unused connection information from the Network Dispatcher database after *FIN-timeout* has elapsed.

**Valid Values:** 0 to 65535

**Default value:** 4000

**FIN-timeout**
Specifies the number of seconds, that a connection has been in the FIN state, after which the executor tries to remove the unused connection information from the Network Dispatcher database.

**Valid Values:** 0 to 65535

**Default value:** 30

**FIN-stale-timer**
Specifies the number of seconds, that a connection has been inactive, after which the executor tries to remove a connection's information from the Network Dispatcher database.

**Valid Values:** 0 to 65535

**Default value:** 1500

**Example:**

```
add cluster
Cluster address [0.0.0.0]? 131.2.24.91
FIN count [4000]?
FIN timeout [30]?
FIN stale timer [1500]?
```

**heartbeat** *address1 address2*
Specifies one path for Heartbeat messages. It is recommended that you configure more than one entry for reliable behavior. The Heartbeat message will flow from *address1*, which belongs to this Network Dispatcher, to *address2*, which belongs to the peer Network Dispatcher.

**address1**
Specifies the IP address of the interface of this Network Dispatcher from which Heartbeat messages will flow.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**address2**
Specifies the IP address of the interface of the peer Network Dispatcher to which Heartbeat messages will flow. This address must be reachable from the interface specified in *address1.*

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**Example:**

```
add heartbeat
Source Heartbeat address [0.0.0.0]? 131.2.25.90
Target Heartbeat Address [0.0.0.0]? 131.2.25.92
```

**port** *cluster-address port# max-weight port-mode*
Specifies the port and port's attributes.

**cluster-address**
Specifies the IP address of the cluster.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**port#** Specifies the port number of the protocol for this cluster.

**Valid Values:** 0 to 65535

**Default value:** 80

**port-mode**
Specifies whether the port will feed all requests from a single client to a single server (known as sticky), use passive ftp (pftp), or use no particular protocols on this cluster (none).

**Valid Values:** sticky, pftp, or none

**Default value:** none

**max-weight**
Specifies the maximum weight for servers on this port. This affects how much difference there can be between the number of requests the executor will give each server.

**Valid Values:** 0 to 100

**Default value:** 20

**Example:**

```
add port
Cluster address [0.0.0.0]? 131.2.25.91
Port number [80]? 80
Max weight (0-100) [20]? 35
Port mode (none=0, sticky=1, pftp=2) [0]?
```

**reach** *address*
Specifies any host address that the Network Dispatcher must be able to reach to run correctly. It can be a server address, a router address, an administration station address or other IP host.

**address**
Specifies the target IP address.

**Valid Values:** Any IP address

**Default value:** 0.0.0.0

**Example:**

```
add reach
Address to reach [0.0.0.0]?
```

**server** *cluster-address port# server-address server-weight server-state*
Specifies the attributes of a server in a cluster.

**cluster-address**
Specifies the IP address of the cluster to which this server belongs.

**Valid Values:** Any IP address

**Default value:** 0.0.0.0

**port#** Specifies the protocol running over the connection to this server.

**Valid Values:** 0 to 65535

**Default value:** 80

**server-address**
Specifies the IP address of the server.

**Valid Values:** Any IP address

**Default value:** 0.0.0.0

**server-weight**
Specifies the weight of the server for the executor. This affects how frequently the Network Dispatcher sends requests to this particular server.

**Valid Values:** 0 to the value of *max-weight* specified on the add port command.

**Default value:** max-weight on port command

**server-state**
Specifies whether the executor should regard the server as available or unavailable when the executor begins processing.

**Valid Values:** 0 (down) or 1 (up)

**Default value:** 1

**Example:**

```
add server
Cluster address [0.0.0.0]? 131.2.25.91
Port number [80]? 80
Server address [0.0.0.0]? 131.2.25.94
Server weight [35]?
Server state (down=0 up=1) [1]?
```

### Parameter Configuration Limits

Table 109 lists the limits for the various items you can configure for a Network Dispatcher.

*Table 109. Parameter Configuration Limits*

| Parameter | Limit |
|---|---|
| Advisors | 32 per 2216 |
| Clusters | 100 per 2216 |
| Heartbeats | 32 per 2216 |
| Ports | 32 per cluster |
| Reachs | 32 per 2216 |
| Servers | 128 per port |

## Clear

Use the **clear** command to clear the entire Network Dispatcher configuration.

**Syntax:**

**clear**

## Disable

Use the **disable** command to disable a Network Dispatcher component.

**Syntax:**

**disable** advisor . . .
backup
executor
manager

**advisor** *name port*

Disables an advisor from the Network Dispatcher.

**name** Specifies the type of advisor.

**Valid values:** 0, 1, 2

0 = FTP
1 = HTTP
2 = MVS

**Default value:** 0

**port** Specifies the port number for this advisor.

**Valid values:** 0 to 65535

**Default value:** 0

**Example:**

```
 disable advisor
Advisor name (0=ftp, 1=http, 2=mvs) [1]? 1
Port number [0]? 80
```

**backup**

Disables the Network Dispatcher's backup function.

**Example:**

```
 disable backup
Backup is now disabled.
```

**executor**

Disables the Network Dispatcher executor. Disabling the executor disables the Network Dispatcher feature.

**Example:**

```
 disable executor
Network dispatcher executor is disabled.
```

**Note:** Disabling the executor will stop the manager, advisors, and the high availability function, if they are currently running.

**manager**

Disables the Network Dispatcher manager. The manager is an optional component. However, if you do not use the manager, the Network Dispatcher will balance the load using a round-robin scheduling method based on the current server weights.

**Example:**

```
 disable manager
Network dispatcher manager is disabled.
```

**Note:** Because the manager component is prerequisite for advisors, disabling the manager will stop all the advisors from running.

# Enable

Use the **enable** command to enable a Network Dispatcher component.

**Syntax:**

**enable** advisor . . .
backup
executor
manager

**advisor** *name port*

Enables an advisor to the Network Dispatcher.

**name** Specifies the type of advisor.

**Valid values:** 0, 1, 2

0 = FTP

1 = HTTP

2 = MVS

**port** Specifies the port number for this advisor.

**Valid values:** 0 to 65535

**Default value:** 0

**Example:**

```
 enable advisor
Advisor name (0=ftp, 1=http, 2=mvs) [1]? 1
Port number [0]? 80
```

**Note:** Because the manager component is a prerequisite for the advisor, you must enable the manager before any advisor can be enabled. You must also set the internal ip address using the **set internal-ip-address** command for the advisor to run correctly. See *Protocol Configuration and Monitoring Reference Volume 1 for Nways Multiprotocol Access Services Version 3 Release 1* for more information about the **set internal-ip-address.** command.

**backup**

Enables the Network Dispatcher's backup function.

**Example:** `enable backup`

**Note:** Before enabling backup, you must add at least one heartbeat

**executor**

Enables the Network Dispatcher executor.

**Example:**

```
enable executor
Network dispatcher executor is enabled.
```

**manager**

Enables the Network Dispatcher manager.

**Example:**

```
enable manager
Network dispatcher manager is enabled.
```

When the manager is enabled for the first time, a manager record is created with the following default values:

**Interval:**
2 seconds

**Refresh-Cycle:**
2

**Sensitivity:**
5 %

**Smoothing:**
1.5

**Proportions:**

**Active:**
50%

**New:** 50%

**Advisor:**
0

**System:**
0

See “Set” on page 789 for a description of the above parameters.

## List

Use the **list** command to display information about the Network Dispatcher.

**Syntax:**

**list** all
advisors
backup
cluster
manager
ports
servers

**all** Displays all Network Dispatcher configuration information. This includes the same information displayed for advisors, backup, cluster, manager, ports, and servers.

**Example:**

```
NDR Config> list all

Executor: Enabled

Manager: Enabled

        Interval        Refresh-Cycle   Sensitivity     Smoothing
        2               2               5 %             1.50
        Proportions:    Active  New     Advisor         System
                        50 %    50 %    0 %             0 %

Advisor:
        Name    Port    Interval        TimeOut         State
        http    80      5               0               Enabled
```

```
        MVS     10007   15              0               Enabled

Backup: Enabled
        Role            Strategy
        PRIMARY         AUTOMATIC

        Reachability:   Address         Mask            Type
                        131.2.25.93     255.255.255.255 HOST
                        131.2.25.94     255.255.255.255 HOST

        HeartBeat Configuration:
           Source Address:  131.2.25.90 Target Address:   131.2.25.92
           Source Address:  132.2.25.90 Target Address:   132.2.25.92

Clusters:
        Cluster-Addr    FIN-count       FIN-timeout     Stale-timer
        131.2.25.91     4000            30              1500

Ports:
        Cluster-Addr    Port#   Weight  Port-Mode
        131.2.25.91     23      20 %    none
        131.2.25.91     80      20 %    none

Servers:
        Cluster-Addr    Port#   Server-Addr     Weight  State
        131.2.25.91     23      131.2.25.93     20 %    up
        131.2.25.91     23      131.2.25.94     20 %    up
        131.2.25.91     80      131.2.25.93     20 %    up
        131.2.25.91     80      131.2.25.94     20 %    up
```

**advisors**
Displays the configuration for the Network Dispatcher advisors.

**backup**
Displays the backup configuration for the Network Dispatcher.

**cluster**
Displays the configuration of the Network Dispatcher clusters.

**manager**
Displays the configuration of the Network Dispatcher manager.

**ports** Displays the configuration of the Network Dispatcher ports.

**servers**
Displays the configuration of the servers associated with the Network Dispatcher clusters.

## Remove

Use the **remove** command to delete part of the Network Dispatcher configuration.

**Syntax:**

**remove**
- advisor . . .
- backup
- cluster . . .
- heartbeat . . .
- port . . .
- reach . . .
- server . . .

**advisor** *name port*
Removes a specific advisor from the Network Dispatcher configuration.

**name** Specifies the type of advisor.

**Valid values:** 0, 1, 2

0 = FTP

1 = HTTP

2 = MVS

**port** Specifies the port number for this advisor.

**Valid values:** 0 to 65535

**Default value:** 0

**Example:**

```
remove advisor
Advisor name (0=ftp, 1=http, 2=mvs) [1]?
Advisor port [0]? 80
```

**backup**

Removes the high availability function.

**Note:** Because backup is a prerequisite for the heartbeat and reach functions removing backup will stop heartbeat and reach from running.

**Example:** `remove backup`

**cluster** *address*

Removes a cluster from the Network Dispatcher configuration.

**address**

Specifies the IP address for the cluster.

**Valid values:** Any valid IP address

**Default value:** 0.0.0.0

**Note:** Removing a cluster address also removes all the ports and servers associated with that cluster.

**Example:**

```
remove cluster
WARNING: Deleting a cluster will make any port or server
         associated with it to also be deleted.
Cluster address [0.0.0.0]? 131.2.25.91
```

**heartbeat** *address*

Removes the heartbeat address from the Network Dispatcher configuration.

**address**

Specifies the IP address for the cluster.

**Valid values:** Any valid IP address

**Default value:** 0.0.0.0

**Example:**

```
remove heartbeat
Target address [0.0.0.0]? 131.2.25.92
```

**port** *cluster-address port#*

Removes a port from a specific cluster in the Network Dispatcher configuration.

**cluster-address**

Specifies the IP address of the cluster.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**port#** Specifies the port number of the protocol for this cluster.

**Valid Values:** 0 to 65535

**Default value:** 0

**Note:** Removing a port will also remove all of the servers associated with that port.

**Example:**

```
remove port
WARNING: Deleting a port will also delete any servers associated with it.
Cluster address [0.0.0.0]? 7.82.142.15
Port number [0]? 80
```

**reach** *address*

Removes a server from the list of hosts the Network Dispatcher must be able to reach.

**address**

Specifies the IP address of the cluster.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**Example:**

```
remove reach
Target address [0.0.0.0]? 9.82.142.15
```

**server** *cluster-address port# server-address*

Removes a server from a cluster and port in the Network Dispatcher configuration.

**cluster-address**

Specifies the IP address of the cluster.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**port#** Specifies the port number of the protocol for this cluster.

**Valid Values:** 0 to 65535

**Default value:** 80

**server-address**

Specifies the IP address of the cluster.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**Example:**

```
remove server
Cluster address [0.0.0.0]? 7.82.142.15
Port number [0]? 80
Server address [0.0.0.0]? 20.21.22.15
```

## Set

Use the **set** command to change the attributes of an existing advisor, cluster, port, or server. You can also define attributes for the Network Dispatcher manager.

**Syntax:**

**set** advisor . . .
cluster . . .
manager . . .
port . . .
server . . .

**advisor** *name port# interval timeout*

Changes the port number, interval, and timeout for an advisor.

**name** Specifies the type of advisor.

0 = FTP
1 = HTTP
2 = MVS

**Valid values:** 0, 1, 2

**Default value:** 1

**port** Specifies the port number for this advisor.

**Valid values:** 0 to 65535

**Default value:** 0

**interval**

Specifies the frequency with which the advisor queries its protocol for each server. After half of this value expires without a response from the server, the adviser considers the protocol unavailable.

**Valid values:** 0 to 65535

**Default value:** 5

**timeout**

Specifies the interval of time, in seconds, after which the advisor considers the protocol unavailable.

To make sure that out-of-date information is not used by the manager in its load-balancing decisions, the manager will not use information from the advisor whose time stamp is older than the time set in this parameter. The advisor timeout should be larger than the advisor polling interval. If the timeout is smaller, the manager will ignore reports that should be used. By default, advisor reports do not time out.

This timeout value typically applies if you disable an advisor. Do not confuse this parameter with the interval/2 timeout previously described, which relates to a server not responding.

**Valid values:** 0 to 65535

**Default value:** 0, which means the protocol is considered always available.

**Example:**

```
set advisor
Advisor name (0=ftp, 1=http, 2=mvs) [0]?
Port number [0]? 21
Interval (seconds) [5]? 10
Timeout (0=unlimited) [0]? 20
```

**cluster** *address FIN-count FIN-timeout FIN-stale-timer*

Changes the FIN-count, FIN-timeout, and FIN-stale-timer for a cluster in the Network Dispatcher configuration.

**address**

Specifies the IP address for the cluster.

**Valid values:** Any valid IP address

**Default value:** 0.0.0.0

**FIN-count**

Specifies the number of connections that must be in FIN state before the executor tries to remove the unused connection information from the Network Dispatcher database after *FIN-timeout* has elapsed.

**Valid Values:** 0 to 65535

**Default value:** 4000

**FIN-timeout**

Specifies the number of seconds after which the executor tries to remove the unused connection information from the Network Dispatcher database.

**Valid Values:** 0 to 65535

**Default value:** 30

**FIN-stale-timer**

Specifies the number of seconds that a connection has been inactive, after which the executor tries to remove a connection's information from the Network Dispatcher database.

**Valid Values:** 0 to 65535

**Default value:** 1500

**Example:**

```
set cluster
Cluster address [0.0.0.0]? 131.2.25.91
FIN count [4000]? 4500
FIN timeout [30]? 40
FIN stale timer [1500]? 2000
```

**manager** *interval proportion refresh sensitivity smoothing*

Sets the values that the manager uses to determine the best server to satisfy a request.

**interval**

Specifies the amount of time, in seconds, after which the manager updates the server weights that the executor uses in load balancing connections.

**Valid values:** 0 to 65535

**Default value:** 2

**proportion**
Specifies the relative importance of external factors in the manager's weighting decisions. The sum of the proportions must equal 100%. The factors are:

**active** The number of active connections on each TCP/IP server as tracked by the executor.

**Valid values:** 0 to 100

**Default value:** 50

**new** The number of new connections on each TCP/IP server as tracked by the executor.

**Valid values:** 0 to 100

**Default value:** 50

**advisor**
Input from the advisors defined to the Network Dispatcher.

**Valid values:** 0 to 100

**Default value:** 0

**system**
Input from the MVS system monitoring tool WLM.

**Valid values:** 0 to 100

**Default value:** 0

**refresh**
Specifies the frequency with which the manager requests status from the executor. This parameter is specified as a number of *intervals*.

**Valid values:** 0 to 100

**Default value:** 2

**sensitivity**
Specifies the percentage weight change for all the servers on a port, after which the manager updates the weights that the executor uses in load balancing connections.

**Valid values:** 0 to 100

**Default value:** 5

**smoothing**
Specifies a limit to the amount that a server's weight can change. Smoothing minimizes the frequency of change in the distribution of requests. A higher smoothing index will cause the weights to change less. A lower smoothing index will cause the weights to change more.

**Valid values:** a decimal value between 1.0 and 42 949 673.00

**Default value:** 1.5

**Note:** You can only specify two places after the decimal point.

**Example:**

```
set manager
Interval (in seconds) [2]? 3
Active proportion [50]? 40
New proportion [50]? 38
Advisor proportion [0]? 20
System proportion [0]? 2
Refresh cycle [2]? 4
Sensitivity threshold [5]? 10
Smoothing index (>1.00) [1.50]? 200
```

**port** *cluster-address port# weight*

Changes the port number and weight for a specific cluster.

**cluster-address**

Specifies the IP address of the cluster.

**Valid Values:** Any IP address.

**Default value:** 0.0.0.0

**port#** Specifies the port number of the protocol for this cluster.

**Valid Values:** 0 to 65535

**Default value:** 80

**weight**

Specifies the weight for servers on this port. This affects how much difference there can be between the number of requests the executor will give each server.

**Valid Values:** 0 to 100

**Default value:** 20

**Example:**

```
set port
Cluster address [0.0.0.0]? 131.2.25.91
Port number [0]? 23
Max. weight (0-100) [20]? 30
```

**server** *cluster-address port# server-address weight state*

Changes the port number, server address, server state, and server weight for a specific server in a cluster.

**cluster-address**

Specifies the IP address of the cluster to which this server belongs.

**Valid Values:** Any IP address

**Default value:** 0.0.0.0

**port#** Specifies the protocol running over the connection to this server.

**Valid Values:** 0 to 65535

**Default value:** 80

**server-address**

Specifies the IP address of the server.

**Valid Values:** Any IP address

**Default value:** 0.0.0.0

**state** Specifies whether the executor should regard the server as available or unavailable when the executor begins processing.

**Valid Values:** 0 (down) or 1 (up)

**Default value:** 1

**weight**

Specifies the weight of the server for the executor. This affects how frequently the Network Dispatcher sends requests to this particular server.

**Valid Values:** 0 to the value of *max-weight* specified on the add port command.

**Default value:** max-weight on port command

**Example:**

```
set server
Cluster address [0.0.0.0]? 131.2.25.91
Port number [0]? 80
Server address [0.0.0.0]? 131.2.25.94
Server weight [20]? 25
Server state (down=0, up=1) [1]? 1
```

## Accessing the Network Dispatcher Monitoring Commands

To access the Network Dispatcher monitoring environment:

1. Enter **talk 5** at the OPCON prompt (*).
2. Enter **feature ndr** at the GWCON prompt (+).

## Network Dispatcher Monitoring Commands

Table 110 summarizes the Network Dispatcher monitoring commands and the rest of the section explains these commands. Enter these commands at the NDR > prompt.

*Table 110. Network Dispatcher Monitoring Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Displays the currently configured attributes of the advisor, clusters, ports, or servers. |
| Quiesce | Specifies that no more connection request should be sent to a server. Also temporarily stops the heartbeat and reach functions. |
| Report | Displays a report of information related to the advisor and the manager. |
| Status | Displays the current status of the counters, clusters, ports, servers, advisor, manager, and backup. |
| Switchover | Forces a Network Dispatcher that is running in standby mode to become the active Network Dispatcher. Use of this command is necessary if you specified manual as the switchover mode. |
| Unquiesce | Allows the Network Dispatcher manager to assign a weight greater than 0 to a previously quiesced server on every port that the server is configured. This action allows new connection requests to flow to the selected server. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

# List

Use the **list** command to display information about the Network Dispatcher.

**Syntax:**

**list** advisor
cluster
port
server

**advisor**
Displays the configuration for the Network Dispatcher advisors.

**Example:**

```
list advisor
Advisor list requested.

-------------------------------------------
| ADVISOR |  PORT  |  TIMEOUT  |  STATUS  |
-------------------------------------------
|  ftp    |    23  |       5   |   ACTIVE |
|  Http   |    80  | unlimited |   ACTIVE |
|  MVS    | 10007  | unlimited |   ACTIVE |
-------------------------------------------
```

**cluster**
Displays the configuration of the Network Dispatcher clusters.

**Example:**

```
list cluster
EXECUTOR INFORMATION:
 ---------------------
Version: 01.01.00.00 - Tue Dec 10 14:15:58 EST 1996
Number of defined clusters: 2

CLUSTER LIST:
-------------
  131.2.25.91
  10.11.12.2
```

**port** Displays the configuration of the Network Dispatcher ports.

**Example:**

```
list port
Cluster Address [0.0.0.0]? 131.2.25.91

-----------------------------------------------
| CLUSTER:       131.2.25.91                  |
|---------------------------------------------|
|   PORT   |    MAXWEIGHT   |  STICKY/PFTP    |
|---------------------------------------------|
|     23   |         30     |    neither      |
|     80   |         20     |    neither      |
-----------------------------------------------
```

**server** Displays the configuration of the servers associated with the Network Dispatcher clusters.

**Example:**

```
list server
Cluster Address [0.0.0.0]? 131.2.25.91

PORT 23 INFORMATION:
 --------------------
Maximum weight.................. 20
Port is sticky.................. FALSE
Port is for passive ftp......... FALSE
All up nodes are weight zero.... FALSE
Total target nodes.............. 2
Currently marked down........... 0
```

```
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1

PORT 80 INFORMATION:
 --------------------
Maximum weight.................. 20
Port is sticky.................. FALSE
Port is for passive ftp......... FALSE
All up nodes are weight zero.... FALSE
Total target nodes.............. 2
Currently marked down........... 0
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1
```

## Quiesce

Use the **quiesce** command to temporarily stop the heartbeat or reach functions or to specify that no more connection requests should be sent to a server.

**Syntax:**

**quiesce** heartbeat
manager
reach

**heartbeat** *address*

Stops the selected path for the heartbeat function. The *address* is the IP address of the remote network dispatcher to which this Network Dispatcher is sending Heartbeat messages.

**Example:**

```
quiesce heartbeat
Remote Address [0.0.0.0]? 131.2.25.94
```

**manager** *address*

Specifies that no more connection requests are to be made to the specified server. *Address* is the IP address of the server.

**Example:**

```
quiesce manager
Server Address [0.0.0.0]? 131.2.25.93
```

**reach** *address*

Stops the Network Dispatcher's polling of the specified address to determine if it is reachable, where *address* is the IP address that is part of the reachability criteria.

**Example:**

```
quiesce reach
Reach Address [0.0.0.0]? 131.2.25.92
```

## Report

Use the **report** command to display a report of the advisor or manager

**Syntax:**

**report** advisor
manager

**advisor** *type port#*

Displays a report of information about a specific advisor.

**type** Is the type of advisor: 0 = ftp, 1 = http, 2 = MVS.

**port#** Is the port number.

**Example:**

```
report advisor
0=ftp, 1=http, 2=MVS
Advisor name [0]? 1
Port number [0]? 80

------------------------------
|      ADVISOR:     http     |
|      PORT:          80     |
------------------------------
| 131.2.25.93      |       0  |
| 131.2.25.94      |      16  |
------------------------------
```

**manager**

Displays a report of the current manager information.

**Example:**

```
report manager

-----------------------------------
|  HOST TABLE LIST  |   STATUS    |
-----------------------------------
|  131.2.25.93      |    ACTIVE   |
|  131.2.25.94      |    ACTIVE   |
-----------------------------------

-------------------------------------------------------------------------------
|131.2.25.91    |WEIGHT | ACTIVE % 50  |  NEW % 50    | PORT %  0 |SYSTEM %  0|
|               ---------------------------------------------------------------|
|PORT:   23     |NOW|NEW| WT | CONNECT | WT | CONNECT | WT | LOAD | WT | LOAD |
-------------------------------------------------------------------------------|
|131.2.25.93    | 10| 10|  10|        0|  10|        0|   0|     0|-999|    -1|
|131.2.25.94    | 10| 10|  10|        0|  10|        0|   0|     0|-999|    -1|
-------------------------------------------------------------------------------|
|PORT TOTALS:   | 20| 20|    |        0|    |        0|    |     0|    |    -2|
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
|131.2.25.91    |WEIGHT | ACTIVE % 50  |  NEW % 50    | PORT %  0 |SYSTEM %  0|
|               ---------------------------------------------------------------|
|PORT:   80     |NOW|NEW| WT | CONNECT | WT | CONNECT | WT | LOAD | WT | LOAD |
-------------------------------------------------------------------------------|
|131.2.25.93    | 10| 10|  10|        0|  10|        1|  16|     0|-999|    -1|
|131.2.25.94    | 10| 10|  10|        0|  10|        1|   3|    16|-999|    -1|
-------------------------------------------------------------------------------|
|PORT TOTALS:   | 20| 20|    |        0|    |        0|    |    16|    |    -2|
-------------------------------------------------------------------------------

-----------------------------------------
| ADVISOR |  PORT  |  TIMEOUT  |  STATUS  |
-----------------------------------------
|  http   |    80  | unlimited |   ACTIVE |
|  MVS    | 10007  | unlimited |   ACTIVE |
-----------------------------------------
Manager report requested.
```

# Status

Use the **status** command to obtain the status of the advisors, backup, counter, clusters, manager, ports, and servers.

**Syntax:**

**status** advisor
backup
cluster
counter

manager

port

server

**advisor** *type port#*

Obtains the status of a specific advisor.

**type** Is the type of advisor. 0 = ftp, 1 = http, 2 = MVS.

**port#** Is the port number.

**Example:**

```
status advisor
0=ftp, 1=http, 2=MVS
Advisor name [0]?
Port number [0]? 21

Advisor ftp on port 21 status:
==============================
Logging level..... 0
Interval.................. 10
```

**backup**

Obtains the status of the backup function.

**Example:**

```
status backup
 Dumping status ...
 Role : PRIMARY   Strategy : AUTOMATIC  State : ND_ACTIVE Sub-State : ND_SYNCHRONIZED
 <<Preferred Target : 132.2.25.92>>

 Dumping HeartBeat Status ...
 .....Heartbeat target : 131.2.25.92 Status : UNREACHABLE
 .....Heartbeat target : 132.2.25.92 Status : REACHABLE

 Dumping Reachability Status ...
 .....Host:131.2.25.93   Local:REACHABLE
 .....Host:131.2.25.94   Local:REACHABLE
```

**cluster** *address*

Obtains the status of a specified cluster, where *address* is the IP address of the cluster.

**Example:**

```
status cluster
Cluster Address [0.0.0.0]? 131.2.25.91

EXECUTOR INFORMATION:
 --------------------
Version: 01.01.00.00 - Tue Dec 10 14:15:58 EST 1996

CLUSTER INFORMATION:
 -------------------
Address........................ 131.2.25.91
Number of target ports......... 2
FIN clean up count............. 4000
Connection FIN timeout......... 30
Active connection stale timer... 1500

PORT 23 INFORMATION:
 -------------------
Maximum weight................. 20
Port is sticky................. FALSE
Port is for passive ftp........ FALSE
All up nodes are weight zero.... FALSE
Total target nodes............. 2
Currently marked down.......... 0
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1

PORT 80 INFORMATION:
 -------------------
Maximum weight................. 20
Port is sticky................. FALSE
Port is for passive ftp........ FALSE
All up nodes are weight zero.... FALSE
```

```
Total target nodes.............. 2
Currently marked down........... 0
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 0 Active: 0 FIN 0 Status: up Saved Weight: -1
```

**counter**

Obtains the status of all counters.

**Example:**

```
status counter
Internal counters from executor:
--------------------------------
Total number of packets into executor...... 2684
Discarded because headers too short........ 0
Packets to non forwarding address.......... 0
Total packets for cluster processing (C)... 2684
Packets not addressed to a cluster(port)... 0

Cluster processing results:
---------------------------
Errors..................................... 0
Discarded.................................. 0
Own address................................ 0
Forward requested.......................... 2684
Forward discarded with error............... 0

Other processing problems:
--------------------------
Total packets dropped (C).................. 0
```

**manager**

Obtains the status of the manager.

**Example:**

```
status manager
Number of defined hosts... 2
Sensitivity.............. 0%
Smoothing factor.......... 2
Interval.................. 3
Weights refresh cycle..... 4

Active connections gauge proportion......... 40%
New connections counter(delta) proportion... 38%
Advisor gauge proportion.................... 20%
System Metric proportion....................  2%

Manager status requested.
```

**port** *clusteraddress port#*

Obtains the status of a specific port, where:

*clusteraddress*
: is the IP address of the cluster.

*port#* is the port number on the cluster.

**Example:**

```
status port
Cluster Address [0.0.0.0]? 131.2.25.91
Port number  [0]? 80

PORT 80 INFORMATION:
 --------------------
Maximum weight.................. 20
Port is sticky.................. FALSE
Port is for passive ftp......... FALSE
All up nodes are weight zero.... FALSE
Total target nodes.............. 2
Currently marked down........... 0
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 12345 Active: 3431 FIN 3780 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 7890  Active: 2980 FIN 2390 Status: up Saved Weight: -1
```

**server** *address*

Obtains the status of a specific server, where *address* is the IP address of the cluster to which the server belongs.

**Example:**

```
status server
Cluster Address [0.0.0.0]? 131.2.25.91

PORT 23 INFORMATION:
 --------------------
Maximum weight.................. 20
Port is sticky.................. FALSE
Port is for passive ftp......... FALSE
All up nodes are weight zero.... FALSE
Total target nodes.............. 2
Currently marked down........... 0
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 140 Active: 50 FIN 45 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 250 Active: 60 FIN 54 Status: up Saved Weight: -1

PORT 80 INFORMATION:
 --------------------
Maximum weight.................. 20
Port is sticky.................. FALSE
Port is for passive ftp......... FALSE
All up nodes are weight zero.... FALSE
Total target nodes.............. 2
Currently marked down........... 0
Servers providing service to this port:
Address: 131.2.25.93 Weight: 20 Count: 12345 Active: 3431 FIN 3780 Status: up Saved Weight: -1
Address: 131.2.25.94 Weight: 20 Count: 7890  Active: 2980 FIN 2390 Status: up Saved Weight: -1
```

## Switchover

Use the **switchover** command to force a Network Dispatcher that is running in standby mode to become the active Network Dispatcher when the switchover strategy is manual. This command must be entered on the host that is running the Network Dispatcher that is in standby mode.

**Syntax:**

**switchover**

## Unquiesce

Use the **unquiesce** command to restart a heartbeat, manager, or reach function that was previously stopped with the **quiesce** command.

Syntax:

**unquiesce** heartbeat
manager
reach

**heartbeat** *address*

Restarts the path for Heartbeat messages, where *address* is the IP address of the remote network dispatcher to which this Network Dispatcher is sending Heartbeat messages.

**Example:**

```
unquiesce heartbeat
Remote Address [0.0.0.0]? 9.10.11.1
```

**manager** *address*

Restarts sending connection requests to the specified server. *Address* is the IP address of the server.

**Example:**

```
unquiesce manager
Server Address [0.0.0.0]? 20.21.22.15
```

**reach** *address*

Restarts the Network Dispatcher's polling of the specified address to determine if it is reachable, where *address* is the IP address that is part of the reachability criteria.

**Example:**

```
unquiesce reach
Reach address [0.0.0.0]? 20.3.4.5
```

# Chapter 66. Using the Data Compression Subsystem

This chapter discusses data compression on a 2216 over Frame Relay and PPP interfaces. It includes these sections:

- "Data Compression Overview"
- "Data Compression Concepts"

Data compression is supported on Frame Relay and PPP interfaces.

## Data Compression Overview

The data compression system provides a means to increase the effective bandwidth of networking interfaces on the device. It is primarily intended for use on slower speed WAN links.

Data compression on the device is supported on PPP and Frame Relay interfaces:

- For PPP interfaces, compression is implemented according to the Compression Control Protocol (CCP) as defined in the Internet Engineering Task Force's RFC 1962. CCP provides the underlying mechanisms by which the use of compression is negotiated and a means for choosing among multiple possible compression algorithms or protocols.

  The device provides two compression protocols: the Stac-LZS protocol, defined in RFC 1974; and the Microsoft Point-to-Point Compression protocol (MPPC), described in RFC 2118. Both of these are based on compression algorithms provided by Stac Electronics.
- For Frame Relay interfaces, compression is implemented according to FRF.9, the *Data Compression over Frame Relay Implementation Agreement* produced by the Frame Relay Forum Technical Committee. FRF.9 describes a Data Compression Protocol (DCP), modeled after PPP's CCP, and similarly provides a means for negotiating various compression algorithms and options. The device supports DCP "mode 1" negotiation. FRF.9 also describes a more generalized "mode 2"; this is not supported. Compression itself is done using the same compression engine as used for the PPP Stac-LZS protocol.

## Data Compression Concepts

Data compression on the device provides a means to increase throughput on network links by making more efficient use of the available bandwidth on a link. The basic principle behind this is simple: represent the data flowing across a link in as compact a manner as possible so that the time needed to transmit it is as low as possible, given a set speed on a link.

Data compression may be performed at many layers in the networking model. At one end of the spectrum, applications may compress data prior to transmitting it to peer applications elsewhere in the network, while at the other end of the spectrum devices may be performing compression at the data link layer, working purely on the bit stream passing between two nodes. How this compression is done and how effective it is depends on a variety of factors, including such things as what network layer the compression is performed at, how much intrinsic knowledge the compressor and decompressor have about the data being compressed, the compression algorithm chosen, and the actual data being compressed. The best

compression can usually be performed at the application layer; for example, a file transfer application usually has the luxury of having an entire file of data available to it prior to attempting compression, and it may be able to try different compression algorithms on the file to see which performs best on that particular file's data. Although this may provide excellent compression for that one type of application, it does little to solve the general problem of compressing the bulk of the traffic flowing over a network, as most networking applications do not currently compress data as they generate it.

Compression on the device takes place at a much lower networking layer, at the data link layer. In the device, compression is performed on the individual packets which are transmitted across a link. The compression is done in real-time as packets flow through the device: the sender compresses a packet just prior to transmitting it, and the decompressor decompresses the packet as soon as it receives it. This operation is transparent to the higher layer networking protocols.

## Data Compression Basics

Data compressors work by recognizing "redundant" information in data, and producing a different set of data which contains as little redundancy as possible. "Redundant" information is any information which can be derived and recreated based on the currently available data. For example, a compressor might function by recognizing repeated character patterns in a data stream and replacing these repeated patterns with a shorter code sequence to represent that pattern. As long as the compressor and decompressor agree on what these code sequences are then the decompressor can always recreate the original data from the compressed data.

This mapping of sequences in the original data to corresponding sequences in the compressed output is commonly called a ***data dictionary***. These dictionaries may be statically defined - experienced-based information available to the compressor and decompressor - or they may be dynamically generated, usually based on the information being compressed. Static dictionaries are most applicable to environments where the data being processed is of a limited, known nature, and not very effective for general-purpose compressors. Most compression systems use dynamic dictionaries, including any compressors used on the device. On a 2216 the data dictionaries are based on the current packet being processed and possibly previously seen packets, but there is no ability to "look ahead" in the data stream as may exist when compression is performed at other layers. For systems where the data dictionary is dynamically derived and based only on previously seen data, the dictionary is also commonly known as a ***history***. The terms history and data dictionary will be used interchangeably throughout the remainder of this chapter, though it should be understood that in other environments a history is a specific form of data dictionary.

The fact that the device uses dynamic dictionaries and that the compressor and decompressor must keep their dictionaries in synchronization means that data compression works on a stream of data passing between two endpoints. Hence, compression on the router is a connection-oriented process, where the endpoints of the connection are the compressor and decompressor themselves. When compression is started on the stream, both ends reset their data dictionaries to some known starting state, and then they update that state as data is received.

Compression could be performed on each individual packet, resetting the histories prior to processing each packet. Normally though, the data dictionaries are not reset between packets, which means that the histories are based not only on the

contents of the current packet, but also the contents of previously seen packets. This usually improves the overall compression effectiveness, because it increases the amount of data which the compressor searches looking for redundancy to remove. As an example, consider the case of one host “pinging” another host with IP: a series of packets is sent out, each one usually nearly identical to the last one sent. The compressor may have little luck compressing the first packet, but it may recognize that each subsequent packet looks very much like the last one sent, and produce highly compressed versions of those packets.

Because the compressor and decompressor histories change with each packet received, the compression mechanisms are sensitive to lost, corrupted, or reordered packets. The compression protocols employed by the device include signalling mechanisms whereby the compressor and decompressor can detect loss of synchronization and resynchronize to each other, such as might be necessary when a packet is lost due to a transmission error. Typically this is done by including a sequence number in each packet which the decompressor will check to make sure it is receiving all packets, in order. If it detects an error, it will reset itself to some known starting state, signal the compressor to do likewise, and then wait (discarding incoming compressed packets) until the compressor acknowledges that it has also reset itself.

Compression on a link typically is performed on data going in both directions over the link. Normally, each end of a connection has both a compressor and decompressor running on it, communicating with their analogs at the other end of the connection, as shown in Figure 58 on page 804. The output (compression) side runs independently of the input (decompression) side. It is possible for completely different compression algorithms to be operating for each direction of the link. When a link connection is established, the compression control protocol for the link will negotiate with the peer to determine the compression algorithm(s) used for the connection. If the two ends cannot agree on compression protocols to use, then no compression will be performed and the link will operate normally - packets will simply be sent in uncompressed form.

### Using Data Compression

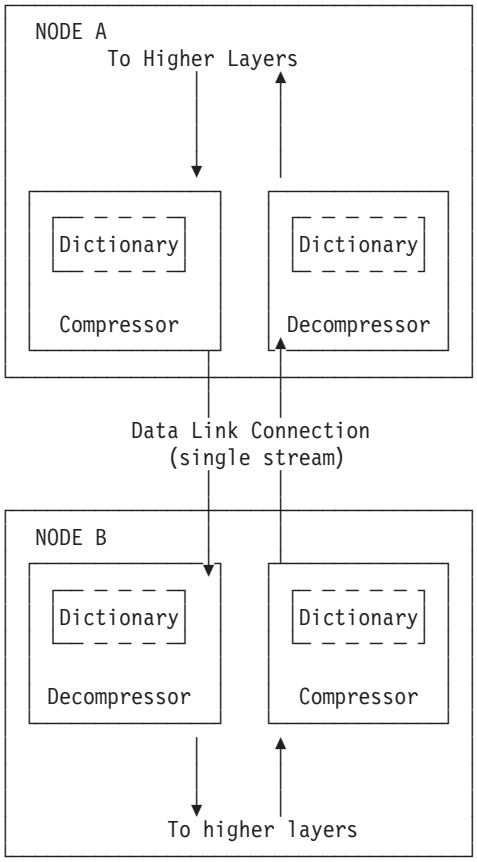



*Figure 58. Example of Bidirectional Data Compression with Data Dictionaries*

A stream really represents a connection between a specific compression process on one end of a link and an associated decompression process on the other end of a link, and thus is more specific than just a “connection” between two nodes; it is possible that a sophisticated compression protocol could split the data flowing between two hosts into multiple streams, compressing each of these streams independently. For example, PPP’s CCP has the ability to negotiate the use of multiple histories over a single PPP link, though the router does not support this.

## Considerations

The choice of whether or not to use data compression is not always an easy one. There are several factors which should be considered before enabling compression on a connection.

### CPU Load

Data compression is a computationally expensive procedure. As the amount of data being compressed increases (per unit time), the more of a load is put on the device’s processor. If the load becomes too great, the performance of the device degrades - on all network interfaces, not just the ones where compression is being performed.

The device actually contains multiple processors and uses asymmetric multiprocessing - for example, link I/O controllers which operate in tandem with the main processor - so the effect of the processor loading is not always readily

measured. Because the compression operation may be overlapped with the transmission of packets, this loading may in fact be totally transparent and pose no problem. Nonetheless, it is possible to overburden the device's processor and degrade performance.

As a general rule of thumb, compression should only be enabled on slow speed WAN links - probably only for links with speeds up to about 64 kilobits per second (the speed of a typical ISDN dial link). The total bandwidth for data being compressed on all links probably should be limited to several hundred kilobits per second. Running compression on all channels of an ISDN Primary Rate adaptor would be unwise.

Some of the device configuration parameters allow you to limit the number of connections which may be concurrently running compression. More interfaces can be enabled for compression then than are actually running it. Once the limit on the number of active compression connections is reached, additional connections will simply not negotiate the use of compression, at least not until an existing compression link shuts down.

## Memory Usage

Another issue to consider when configuring compression is the memory requirement. Compression and decompression histories occupy a fair amount of memory, which is a limited resource in the device. The Stac-LZS algorithm for example requires about 16 Kbytes for a compression history, and about 8 Kbytes for a decompression history. This problem is magnified by the fact that these histories must exist for each connection which is established: a compression history is synchronized with a corresponding decompression history in a peer router. For a PPP link, this implies one compression history and one decompression history (assuming that data compression is running bidirectionally on the link). On a Frame Relay link, there could be many such histories required, one pair for each virtual connection (DLCI) which is established.

The device allocates a limited number of compression and decompression histories when it boots. These are always allocated in pairs known as ***compression contexts*** - a context is simply one compression history coupled with one decompression history. Technically, compression and decompression are independent functions and the allocation of compression and decompression histories could be performed independently; however, in practice compression is almost always run bidirectionally and so memory is managed and configured in terms of contexts rather than individual histories as a way of simplifying operation. Each context is allocated 24 Kbytes which includes the memory required for compression and decompression histories.

Whenever the device attempts to establish a compression connection on a link, it begins by reserving a context from the allocated pool of contexts. If no contexts are available, then compression is not performed on that connection. The router may attempt to start compression on that connection later as contexts become available.

The number of compression contexts which are allocated is a configurable parameter. Setting the number of contexts allocated limits both the amount of memory used and the maximum number of connections which may be simultaneously operating with compression. Limiting the number of simultaneously operating compression connections provides a means to help control the CPU loading problem.

### Data Content

The actual nature of the data being transmitted on a connection should be considered before enabling compression for that connection. Compression works better on some types of data than others. Packets which contain a lot of nearly identical information - for example a set of packets generated from an IP “ping” - will normally compress extremely well. A typical assortment of random text and binary data going over a link will usually compress in ratios around 1.5:1 to 3:1. Some data simply will not compress well at all. In particular, data which has already been compressed will seldom compress further. In fact, data which has been previously compressed may actually expand when fed through the compression engine.

If it is known in advance that most of the data flowing over a connection will consist of compressed data, then it is recommended that compression not be enabled for that connection. An example where this might occur is a connection to a host which was set up to be primarily a FTP file archive site, where all the files available to be transferred are stored in compressed form on the host.

### Link Layer Compression

A final factor to consider is the nature of the network link between the two hosts. Compression could be performed at a lower layer than even the device’s hardware interfaces. In particular, many modern modems incorporate data compression mechanisms in their hardware and firmware. If compression is being performed on the link at a lower layer (outside the device), then it is best not to enable data compression on the device for that interface. As already mentioned, compressing an already compressed data stream is normally ineffective, and in fact may degrade performance slightly. Unless there is some particular reason to believe that the router will do a much better job of compression than the link hardware, it is best to let the link hardware do the compression.

## Using Data Compression on PPP Links

The 2216 uses the PPP Compression Control Protocol (CCP) to negotiate the use of compression on a link. CCP provides a generalized mechanism to negotiate the use of a particular compression protocol, possibly even using a different protocol in each direction of the link, and various protocol-specific options. The software supports the Stac-LZS and MPPC protocols, so the peer must also provide support for at least one of these algorithms to successfully negotiate data compression between the two nodes. The two nodes must also agree on the algorithm-specific options for compression to operate.

## Configuring Data Compression on PPP Links

To configure data compression on PPP links:

1. Enable the CCP protocol on the link with the **enable ccp** command. This enables the link to negotiate compression with the other node. Negotiation includes what compression protocol to use and any protocol-specific options.
2. Select which compression protocols may be negotiated using the **set ccp protocols** command.
3. Set the negotiable parameters for each compression protocol using the **set ccp options** command.

You can display the current compression configuration using the **list ccp** command.

Table 111 lists the available commands and Figure 59 is an example of configuring compression on a PPP link. For detailed description of these commands, see “Point-to-Point Configuration Commands” on page 526.

*Table 111. PPP Data Compression Configuration Commands*

| Data Compression Command | Action |
|---|---|
| disable ccp | Disables data compression. |
| enable ccp | Enables data compression. |
| set ccp options | Sets options for the compression algorithm. |
| set ccp algorithms | Specifies a prioritized list of compression protocols. |
| list ccp | Displays compression configuration. |

```
Config> network 1

Point-to-Point user configuration
PPP Config> enable ccp
PPP Config> set ccp options2
STAC: # histories [1]? 1
STAC: check mode (0=none, 1=LCB, 2=CRC, 3=Seq, 4=Ext) [3]? 3
PPP Config> list ccp
CCP Options
-----------

Data Compression enabled
Algorithm list: STAC-LZS
Stac: histories 1
Stac: check_mode SEQ
```

*Figure 59. Example of Configuring Compression on a PPP Link*

**Notes:**

1. The network command selects the network interface for the PPP link. If the link is a PPP dial circuit, you must then use the **encapsulator** command to access the PPP configuration menu.
2. If you enable CCP and do not set protocols for the link, the software automatically sets the link to use protocols STAC and MPPC as if you had entered the command **set ccp protocols stac mppc**.

   If you set multiple protocols, the order of the protocols determines the negotiation preference for the link.
3. If you enter **set ccp protocols none**, the software will automatically disable compression on the link.

## Monitoring Compression on PPP Links

You monitor compression as you would other PPP components. “Accessing the Interface Monitoring Process” on page 541 describes how to access the PPP console environment and details about the commands. Table 112 lists the compression-related commands. Figure 60 on page 808 shows an example of listing compression on a PPP interface.

*Table 112. PPP Data Compression Monitoring Commands*

| Command | Function |
|---|---|
| **list control ccp** | Lists CCP state and negotiated options. |
| **list ccp** | Lists CCP packet statistics. |
| **list cdp** or **list compression** | Lists compressed datagram statistics. |

```
+ network 1
PPP > list control ccp

CCP State:                Open
Previous State:           Ack Sent
Time Since Change:        2 minutes and 52 seconds

Compressor:   STAC-LZS histories 1, check_mode SEQ
Decompressor: STAC-LZS histories 1, check_mode SEQ

PPP > list ccp

CCP Statistic             In                        Out
-------------             --                        ---

Packets:                  2                         3
Octets:                   18                        27
Reset Reqs:               0                         0
Reset Acks:               0                         0
Prot Rejects:             1                         -

PPP > list cdp

Compression Statistic     In                        Out
---------------------     --                        ---

Packets:                  19541                     19542
Octets:                   2550673                   2740593
Compressed Octets:        821671                    899446
Incompressible Packets:   0                         0
Discarded Packets:        0                         -
Prot Rejects:             0                         -
Compression Ratios:       3.11                      3.24
```

*Figure 60. Monitoring Compression on a PPP Interface*

## Using Data Compression on Frame Relay Links

After configuring the global compression parameters and enabling compression on the interface, you must then set the parameters for each individual circuit (PVC) on the Frame Relay interface. Each circuit defined for the interface may have compression enabled on the circuit, and each circuit which successfully negotiates the use of compression uses one compression context from the global pool. You can also disable compression on the interface which means none of the circuits on that interface will be eligible to carry compressed data traffic.

## Configuring Data Compression on Frame Relay Links

To configure data compression on FR links:

1. Enable compression on the interface using the **enable compression** command. This enables the link to negotiate compression with the other node.
2. Enable compression on each new PVC that will carry compressed data with the **add permanent-virtual-circuit** command. You can change existing PVCs using the **change permanent-virtual-circuit** command.

You can display the current compression configuration using the **list lmi** or **list permanent-virtual-circuit** commands.

Table 113 on page 809 lists the commands available for configuring compression on a Frame Relay link and Figure 61 on page 809 is an example of configuring a Frame Relay Link. See “Frame Relay Configuration Commands” on page 475 for details about the Frame Relay configuration commands.

```
Config> net 2

Frame Relay user configuration

FR Config> enable compression
Maximum number of run-time compression PVCs (zero means no limit) [0]? 0
Do you want orphan PVCs to perform compression [Y]? n
The number of currently defined non-compression PVCs is 4
Would you like to change them all to compression PVCs [N]? y

FR Config> add perm

Circuit number [16]? 22
Committed Information Rate (CIR) in bps [65536]?
Committed Burst Size (Bc) in bits [64000]?
Excess Burst Size (Be) in bits [0]?
Assign circuit name []? cir22
Is circuit required for interface operation [N]?
Do you want to have data compression performed [Y]?

FR Config>list lmi

                        Frame Relay Configuration


  LMI enabled            =      No  LMI DLCI                      =       0
  LMI type               =    ANSI  LMI Orphans OK                =     Yes
  CLLM enabled           =      No  Timer Ty seconds              =      11

  Protocol broadcast     =     Yes  Congestion monitoring         =     Yes
  Emulate multicast      =     Yes  CIR monitoring                =      No
  Notify FECN source     =      No  Throttle transmit on FECN     =      No

  Data compression       =     Yes  Orphan compression            =      No
  Compression PVC limit  =    None  Number of compression PVCs    =       2

  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
  Default CIR            =   65536  Default Burst Size            =   64000
  Default Excess Burst   =       0

FR Config>list perm

  Maximum PVCs allowable =      64
  Total PVCs configured  =       2

             Circuit                Circuit   Circuit     CIR    Burst  Excess
              Name                  Number     Type      in bps  Size   Burst
---------------------------------- ------- ----------- ------- ------- -----
 circ16                              16    @ Permanent   65536   64000       0
 cir22                               22    @ Permanent   65536   64000       0

* = circuit is required
# = circuit is required and belongs to a required PVC group
@ = circuit is data compression capable
```

*Figure 61. Example of Configuring Compression on a Frame Relay Link*

*Table 113. Data Compression Configuration Commands*

| Command | Action |
|---|---|
| **add permanent-virtual-circuit** *#* | Use to enable data compression on a specific PVC defined on an interface. |
| **change permanent-virtual-circuit** *#* | Use to change whether a specific PVC will compress data. |
| **disable compression** | Disables data compression. |
| **enable compression** | Enables data compression. |
| **list lmi** | Displays the current configuration of the interface. |

*Table 113. Data Compression Configuration Commands (continued)*

| Command | Action |
|---|---|
| **list permanent** | Lists summary information about circuits. |

**Note:** Enabling compression on orphan circuits will decrease the number of available compression contexts available for the native PVCs on the device.

If you enable compression on a Frame Relay interface, that already has compression enabled, the software asks you if you want to change compression parameters on the interface as shown in 810. You can change compression on the interface without disabling compression.

```
Example of changing compression on Frame Relay Interfaces
Config> net 2

Frame Relay user configuration

FR Config> enable compression
Data compression already enabled.
Do you wish to continue and change an interface parameter [Y]
Maximum number of run-time compression PVCs (zero means no limit) [0]? 32
Do you want orphan circuits to perform compression []?
Do you want to change the compression capability of all of your existing PVCs [N]?
```

## Monitoring Data Compression on Frame Relay Links

You monitor compression as you would other Frame Relay components. “Frame Relay Monitoring Commands” on page 498 describes how to access the Frame Relay console environment and details about the commands. Table 114 lists the compression-related commands. Figure 62 on page 811 shows an example of listing compression on a Frame Relay interface.

*Table 114. Frame Relay Data Compression Monitoring Commands*

| Command | Display |
|---|---|
| **list lmi** | Lists the current status of the interface. |
| **list permanent** | Lists summary information about circuits. |
| **list circuit** | Lists the current status of a circuit. |

```
+ network 2
FR 2 > list lmi

Management Status:
  ------------------

    LMI enabled          =      No  LMI DLCI                   =       0
    LMI type             =    ANSI  LMI Orphans OK             =     Yes
    CLLM enabled         =      No

    Protocol broadcast   =     Yes  Congestion monitoring      =     Yes
    Emulate multicast    =     Yes  CIR monitoring             =      No
    Notify FECN source   =      No  Throttle transmit on FECN  =      No
    PVCs P1 allowed      =      64  Interface down if no PVCs  =      No
    Line speed (bps)     =   64000  Maximum frame size         =    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
    Default CIR          =   65536  Default Burst Size         =   64000
    Default Excess Burst =       0

    Current receive sequence  =       0
    Current transmit sequence =       0
    Total status enquiries    =       0  Total status responses   =       0
    Total sequence requests   =       0  Total responses          =       0

    Data compression enabled  =     Yes  Orphan Compression       =      No

    Compression PVC limit     =    None  Active compression PVCs  =       1

  PVC Status:
  -----------

    Total allowed  =      64  Total configured  =       1
    Total active   =       1  Total congested   =       0
    Total left net =       0  Total join net    =       0

FR 2 > list permanent

 Circuit                                  Orphan  Type/   Frames      Frames
 Number            Circuit Name           Circuit State Transmitted  Received
 ------- -------------------------------- ------- ----- ----------- --------

     16  circ16                             No   @ P/A        58364      58355
     22  circ22                             No   & P/A        58364      58355

    A - Active   I - Inactive   R - Removed   P - Permanent   C - Congested
    * - Required               # - Required and belongs to a PVC group
    @ - Data compression capable but not operational
    & - Data compression capable and operational
```

*Figure 62. Monitoring Compression on a Frame Relay Interface or Circuit (Part 1 of 2)*

## Using Data Compression

```
FR 2 > list circuit 22

 Circuit name = circ22

     Circuit state        =    Active  Circuit is orphan   =           No
     Frames transmitted   =     58391  Bytes transmitted   =      2676894
     Frames received      =     58383  Bytes received      =      2671009
     Total FECNs          =         0  Total BECNs         =            0
     Times congested      =         0  Times Inactive      =            0
     CIR in bits/second   =     65536  Potential Info Rate =        64000
     Committed Burst (Bc) =     64000  Excess Burst (Be)   =            0
     Minimum Info Rate    =     16000  Maximum Info Rate   =        64000
     Required             =        No  PVC group name      =   Unassigned

     Compression capable  =       Yes  Operational         =          Yes
     R-R's received       =         0  R-R's transmitted   =            0
     R-A's received       =         0  R-A's transmitted   =            0
     R-R mode discards    =         0  Enlarged frames     =            0
     Decompress discards  =         0  Compression errors  =            0
     Rcv error discards   =         0

     Compression ratio    =  1.00 to 1  Decompression ratio =    1.00 to 1

     Current number of xmit frames queued                  =            0
     Xmit frames dropped due to queue overflow             =            0
```

*Figure 62. Monitoring Compression on a Frame Relay Interface or Circuit (Part 2 of 2)*

# Chapter 67. Configuring and Monitoring Data Compression

Configuring data compression on a 2216 is a two-step process. The core compression system is a “Feature” in the software. You set and monitor global parameters by selecting the CMPRS feature in the Configuration and Monitoring tasks (the GWCON and CONFIG processes in the router). In addition to configuring the global parameters, you must also configure compression for each network interface (PPP or Frame Relay) on which you will transmit compressed data traffic.

This section describes configuring and monitoring the compression feature first and then describes configuring and monitoring compression on PPP and Frame Relay interfaces.

## Configuring the Compression Feature

The only configurable parameter for the compression feature is the number of compression contexts to allocate when the device boots. The number of available contexts limits the number of connections that can be active simultaneously, as well as determining the amount of memory set aside for compression histories. Setting the number of contexts to zero disables compression on all interfaces.

In the Config process, enter **feature cmprs** at the `Config >` prompt to access the compression configuration commands. To change the number of contexts allocated, use the **SET MAXCONTEXTS *n*** command where ***n*** is the number of contexts. To see the current configuration, use the **list** command. The complete set of configuration commands is summarized in Table 115 on page 814, and a configuration example is shown in Figure 63 on page 814.

```
Config> feature cmprs

Data Compression Global Configuration
CMPRS Config> ?
LIST
SET
EXIT

CMPRS Config> set ?
MAXCONTEXTS

CMPRS Config> set maxcontexts
Number of compression contexts to allocate? (0 - 1000) [0]? 10

CMPRS Config> list
Number of compression contexts to allocate: 10
```

*Figure 63. Configuring the Compression Feature*

*Table 115. Compression Configuration Commands*

| Command | Action |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Displays the current setting of maxcontexts. |
| Set | Sets the maximum number of compression contexts available for all interfaces. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to display the current setting of *maxcontexts*.

**Syntax:**

**list**

## Set

Use the **set** command to set the maximum number of interfaces that can use data compression simultaneously.

**Syntax:**

**set** maxcontexts *n*

**maxcontexts** *n*

Sets the maximum number of compression contexts available for the interfaces. This parameter causes the device to allocate a pool of memory for compression contexts. Setting maxcontexts to 0 prevents any interface from compressing data even if you enabled compression on the interface.

**Note:** Setting this value too high can result in excessive memory use and decreased throughput for the device.

**Default Value:** 0

**Valid Values:** 0 to 1000

**Example:** `set maxcontexts`

```
Number of compression contexts to allocate? (0-1000)? [0]? 10
```

## Monitoring the Compression Feature

In the monitoring process, enter **feature cmprs** at the + prompt to access the compression monitoring commands. Table 116 lists the available commands.

*Table 116. Compression Monitoring Command*

| Command | Action |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | List either the memory or contexts in use. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to list either the memory or the contexts currently in use.

**Syntax:**

**list** all
contexts usage
memory usage

**all** Displays the contexts in use and the interfaces using the contexts, and the memory usage statistics. The output is a combination of list contexts usage and list memory usage displays.

**Example: `list all`**

**context usage**
Displays all of the compression contexts currently allocated by an interface. This display allows you to see which interfaces are currently compressing data traffic

**Example: `list context usage`**

```
Compression System Context (Data Dictionary) Usage
--------------------------------------------------

    CTX  Net Interface   Channel   Status
    ---  --- ----------  --------- -------------------
      0    2  FR/0              16 In use
      1    1  PPP/0              1 In use

Total: 10     Free: 8     In Use/Reserved: 2
```

**CTX** This is the context number, which is an identifying tag for the context. The device creates a pool of contexts when it boots, and assigns a number to each context in the pool. The context number is also displayed in some of the compression-related ELS messages.

**Net** This is the number of the network interface which has allocated a particular context.

**Interface**
This is the name of the network interface.

**Channel**

The channel is an identifier used to distinguish between multiple contexts allocated to the same network interface. The network number and channel number together uniquely identify a single compression stream. For PPP links, only a single compressed data stream runs on the link, and this number will always be 1. For Frame Relay links, this number is the virtual circuit number (DLCI) of the particular circuit that is carrying compressed traffic.

**Status**

This field indicates the current status of the context, which will almost always be “In use”. Occasionally “Defunct” may appear which indicates that compression has been shut down on a link, but that the context has not yet been released to the pool for reuse.

**memory usage**

Displays basic statistics about the current state of the compression feature. The output shows the number of compression contexts which have been allocated, the number of contexts currently in use, the amount of memory required by a context, and the total amount of memory reserved for compression contexts.

**Example:**

```
list memory usage

Compression System Memory Usage Statistics
------------------------------------------

Number of contexts allocated:            0 *        in use: 0
Size of compression context:             24624
   = Max compression history size:       16396
   + Max decompression history size:     8200
   + Overhead:                           28
Total memory allocated for contexts:     0

* Compression is disabled due to inability to allocate
  the requested number of contexts (500).
```

# Chapter 68. Using Local or Remote Authentication

Authentication is the action of determining who a user (or entity) is. Authenticating user access for the PPP protocol on the 2216 extends the flexibility of user profile management as it relates to PPP authentication protocols PAP and CHAP. See “PPP Authentication Protocols” on page 517 for additional information about configuring PAP and CHAP.

Authentication can be configured locally or can be configured to consolidate user configuration by using authentication servers that are available on the network to service authentication requests for the entire network. The IBM 2216 implements locally maintained authentication as well as the following authentication server protocols:

- Radius
- TACACS
- TACACS+

## Using Authentication, Authorization, and Accounting (AAA) Security

Authentication, Authorization, and Accounting (AAA) Security are configurable protocols that allow you to control access to your services. AAA can be configured to be performed for local or remote .

A security protocol can be configured for three types of functions.

- PPP links
- Login users (Telnet/Console Login)
- Tunnels

The configuring is done by setting a primary and secondary server. The server information is configured and stored separately from the AAA configuration. You reference a server profile by a name provided at configuration time.

Under all circumstances accounting cannot be done locally and must be either Radius or TACACS+.

Authorization can only be done locally or through remote authentication using Radius or TACACS+.

## What is AAA Security

AAA Security is the name of the security system for this device. It includes:

**Authentication**
The action of identifying a user. Authentication utilizes a name and a password for access.

**Authorization**
The action of determining what a user is allowed to access. An authorization request might indicate that the user is not authenticated. The authorization agent then determines if an unauthenticated user is allowed to access the services in question.

**Accounting**
The action of recording when a user has started or stopped a session. There are two types of accounting records supported.

**Start records**
Indicates that a service is about to begin.

**Stop records**
Indicates that a service has been terminated.

## Using PPP

For the Point-to-Point Protocol (PPP) you can configure the following:

- Authentication
- Authorization
- Accounting

Each function can have its own security protocol and is independently configured.

- Setting the authentication protocol will have no effect on authorization or accounting.
- Setting the authorization protocol will have no effect on authentication or accounting.
- Setting the accounting protocol will have no effect on authentication or authorization.
- Setting AAA to remote will set authentication to remote, authorization to remote and set accounting to remote.
- Setting AAA to local will set authentication to local, authorization to local and set accounting to ignore. Disabling authentication or authorization is not allowed.

See “Point-to-Point Configuration Commands” on page 526 for details about the PPP configuration commands that you use in this environment.

## Valid PPP Security Protocols:

The following are valid PPP security protocols:

**Authent Method**
Local, RADIUS, TACACS Plus, TACACS

**Authorization**
Local, RADIUS, TACACS Plus

**Accounting Method**
RADIUS, TACACS Plus

*Table 117. Set PPP Security Protocols*

| Action | Authent | Author | Acct |
|---|---|---|---|
| set AAA local | local | local | ignore |
| set AAA remote | remote | remote | remote |
| set AUTHENT local | local | ignore | ignore |
| set Author local | ignore | local | ignore |
| set AUTHENT to remote | remote | ignore | ignore |
| set AUTHOR to remote | ignore | remote | ignore |

*Table 117. Set PPP Security Protocols (continued)*

| Action | Authent | Author | Acct |
|---|---|---|---|
| set ACCOUNTING to remote | ignore | ignore | remote |
| disable ACCOUNTING | ignore | ignore | disabled |
| disable AUTHENT | n/a | n/a | n/a |
| disable AUTHOR | n/a | n/a | n/a |

## Using Login

For a Login AAA configuration, either Remote or Local can be selected. If Local authentication is desired, then Local authorization must also be used. If Remote authentication is selected, then, Remote authorization must be used. Accounting is not supported locally, so when authenticating and authorizing locally then accounting must be disabled.

**Attention:** Before enabling console login, save the configuration with console login disabled. If login authentication is set to a remote server using Radius or TACACS+ and the router is unable to reach the authentication server, then access to the router is denied. By disabling the console login, a lockout situation is prevented.

When Remote authentication is configured then authorization can be set to another remote authorization protocol Radius or TACACS+, and accounting can be set to use Radius or TACACS+.

- Setting AAA to local will set authentication to local, authorization to local, and accounting to disabled.
- Setting AAA to remote will set authentication to remote, authorization to same as authentication, and accounting to same.
- Setting the authentication protocol to local will automatically set the authorization protocol to same and disable accounting.
- Setting the authentication protocol to remote will automatically set the authorization protocol to same if the authorization protocol is set to local, ignore the accounting protocol.
- Setting the authorization protocol to remote will automatically set the authentication protocol to the same if the authentication protocol is set to local, ignore accounting protocol.
- Setting the accounting protocol to remote will automatically set authentication protocol to the same if the authentication protocol is set to local, and set the authorization protocol to the same if authorization is set to local.
- Setting the accounting protocol to disable will have no effect on the authentication or authorization protocol.
- Disabling authentication or authorization is not allowed.

## Valid Login/Admin Security Protocols

The following are valid Login/Admin security protocols.

**Authent/Author**
Local, RADIUS, TACACS Plus

**Accounting Method**
RADIUS, TACACS Plus

### Using Local or Remote Authentication

*Table 118. Set Login Security Protocols*

| Action | Authent | Author | Acct |
|---|---|---|---|
| set AAA local | local | local | disabled |
| set AAA remote | remote | remote | remote |
| set AUTHENT local | local | local | disabled |
| set AUTHOR local | local | local | disabled |
| set AUTHENT to remote | remote | remote, if local else ignore | ignore |
| set AUTHOR to remote | remote, if local else ignore | remote | ignore |
| set ACCOUNTING to remote | remote, if local else ignore | remote, if local else ignore | remote |
| disable ACCOUNTING | ignore | ignore | disabled |
| disable AUTHEN | n/a | a | n/a |
| disable AUTHOR | n/a | n/a | n/a |

## Using Tunnels

Tunnel authentication must be set to the same as tunnel authorization. When Tunnel authentication is set to either Local or Remote then Accounting may be enabled. The Tunnel authorization and authentication server must be the same.

## Valid Tunnel Security Protocols

The following are valid Tunnel security protocols:

**Authent/Author**
Local, RADIUS

**Authorization**
Local, RADIUS

**Accounting Method**
RADIUS, TACACS Plus

*Table 119. Set Tunnel Security Protocols*

| Action | Authent | Author | Acct |
|---|---|---|---|
| set AAA local | local | local | ignore |
| set AAA remote | remote | remote | remote |
| set AUTHENT local | local | local | ignore |
| set Author local | local | local | ignore |
| set AUTHENT to remote | remote | remote | ignore |
| set AUTHOR to remote | remote | remote | ignore |
| set ACCOUNTING to remote | ignore | ignore | remote |
| disable ACCOUNTING | ignore | ignore | disabled |
| disable AUTHENT | n/a | n/a | n/a |
| disable AUTHOR | n/a | n/a | n/a |

## Understanding Authentication Servers

An ***authentication server*** is a server in the network that validates userids and passwords for the network. If a device is configured for authentication through an authentication server and the device receives a packet from an authentication protocol, the device passes a userid and password to the server for authentication. If the userid and password are correct, the server responds positively. The device can then communicate with the originator of the request. If the server does not find the userid and password it receives from the device, it responds negatively to the device. The device then rejects the session from which it got the authentication request.

# Chapter 69. Configuring Authentication

This chapter describes the configuration and operational commands for authentication. It includes the following sections:

- "Accessing the Authentication Configuration Prompt"
- "Authentication Configuration Commands"

## Accessing the Authentication Configuration Prompt

To access the `Authent config >` prompt:

1. Enter **talk 6** at the `*` prompt.
2. Enter **feature auth** at the `Config >` prompt.

## Authentication Configuration Commands

Table 120 lists the commands available at the `Authent config >` prompt.

*Table 120. Authentication Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Disable | Disables accounting for AAA. |
| List | Displays the AAA configuration parameters. |
| Login | Configures AAA for login. |
| Nets-info | Displays information about local PPP authentication. |
| Password-rules | Configures password rules (enables or disables). |
| PPP | Configures AAA for PPP. |
| Quickset | Configures the authentication method quickly. |
| Servers | Configures individual remote AAA servers. |
| Set | Configures Authentication parameters regardless of type. |
| Tunnel | Configures AAA for L2TP tunnels. |
| User-profile | Configures local PPP users. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Disable

Use the **disable** command to disable accounting.

**Syntax:**

**disable** accounting

## List

Use the **list** command to display the AAA parameters.

**Syntax:**

**list** accounting
authentication
authorization
all
config

```
AAA Config> list all
ppp AAA configuration...
 ppp authentication        : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
 ppp authorization         : locallist
 ppp accounting            : Disabled
tunnel AAA configuration...
 tunnel authentication     : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
 tunnel authorization      : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
 tunnel accounting         : Disabled
login AAA configuration...
 login authentication      : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
 login authorization       : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
 login accounting          : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>

AAA Config> list accounting all
accounting AAA configuration...
 accounting ppp            : Disabled
 accounting tunnel         : Disabled
 accounting login          : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
```

```
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
AAA Config> list accounting config
accounting ppp            : Disabled
accounting login          : Radius        serv01
accounting tunnel         : Disabled

AAA Config> list authentication all
authentication AAA configuration...
 authentication ppp        : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
 authentication tunnel     : Radius        serv01
   authorizeAuthent          YES
   Primary server address    1.1.1.1
   Secondary server address  2.2.2.2
     Request tries           3
     Request interval        3
   Key for encryption        <notSet>
```

# Login

Use the **login** command to configure AAA for login.

Table 121 lists the subcommands available with the **login** command.

*Table 121. Login Subcommands*

| Command | Function |
|---|---|
| Disable | Disables accounting for login. |
| List | Displays the AAA configuration parameters for login. |
| Set | Sets the AAA configuration parameters for login. |

## Disable

Use the **login disable** command to disable accounting.

**Syntax:**

**login disable** accounting

## List

Use the **login list** command to display the AAA configuration parameters.

**Syntax:**

**login list** all
accounting
authentication
authorization
config

### Set

Use the **login set** command to configure authentication parameters.

**Syntax:**

**login set** aaa
accounting
authentication
authorization

**aaa** *authtype*
Sets the authentication, authorization, and accounting type. *Authtype* is one of the following:

**local** Sets the authentication, authorization, and accounting type to use a locally-maintained user database.

**remote**
Sets the authentication, authorization, and accounting type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**accounting** *authtype*
Sets the accounting type. *Authtype* is one of the following:

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authentication** *authtype*
Sets the authentication type. *Authtype* is one of the following:

**local** Sets the authentication type to use a locally-maintained user database.

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authorization** *authtype*
Sets the authorization type. *Authtype* is one of the following:

**local** Sets the authorization type to use a locally-maintained user database.

**remote**
Sets the authorization type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

## Nets-info

Use the **nets-info** command to display the currently configured PPP authentication protocol on each PPP interface.

**Syntax:**

**nets-info**

## Password-rules

Use the **password-rules** command to configure the password (enable or disable).

Table 122 lists the subcommands available with the **password-rules** command.

*Table 122. Login Subcommands*

| Command | Function |
|---|---|
| Disable | Disables a password rule. |
| Enable | Enables a password rule. |
| List | Displays the current state of the password rules (enabled or disabled). |

### Disable

Use the **password-rules disable** command to disable any or all of the password rules.

**Syntax:**

**password-rules disable** all
compare-ident-prev
change-days
first-non-numeric
force-change
ident-chars
last-non-numeric
lockout
minimum-length
one-alpha
one-nonalpha
prev-three
userid-contained

**compare-ident-prev**
Compares the previous user identity with the user requesting a password change.

**change-days**
The maximum number of days before a password change is required.

**Valid values:** 0 to 360

**Default value:** 180

**first_non-numeric**
The first character of a password cannot be numeric.

**Valid values:** any non-numeric character

**Default value:** none

**force-change**
Forces a password change after the maximum change-days has expired. You are prompted for the old password, new password and to verify the new password.

**Valid values:** 0 to 360

**Default value:** 180

**ident-chars**
Cannot contain more than 3 characters used in a previous password in the same position.

**last-non-numeric**
The last character in the password cannot be numeric.

**Valid values:** any non-numeric character

**Default value:** none

**lockout**
The number of times you can try a password before you are locked out.

**Valid values:** 0 to 360

**Default value:** 3

**minimum-length**
The least number of characters required to have a valid password.

**Valid values:** 1 to 31

**Default value:** 8

**maximum-length**
The maximum number of characters a password can contain.

**Valid values:** 1 to 31

**Default value:** 8

**one-alpha**
At least one character in the password must be an alpha.

**one-nonalpha**
At least one character in the password must be numeric.

**prev-three**
The password cannot be the same as any of the last three passwords.

**userid-contained**
The password cannot contain the userid as a part of the password.

## Enable

Use the **password-rules enable** command to enable any or all of the password rules. See the **disable** command for a list of password rule descriptions.

**Syntax:**

**password-rules enable** all
compare-ident-prev
change-days

first-non-numeric
force-change
ident-chars
last-non-numeric
lockout
minimum-length
one-alpha
one-nonalpha
prev-three
userid-contained

### List

Use the **password-rules list** command to display the current state of the password rules (disabled or enabled).

**Syntax:**

**password-rules list**

## PPP

Use the **ppp** command to configure AAA for PPP.

Table 123 lists the subcommands available with the **ppp** command.

*Table 123. PPP Subcommands*

| Command | Function |
|---|---|
| Disable | Disables accounting for PPP. |
| List | Displays the AAA configuration parameters for PPP. |
| Set | Sets the AAA configuration parameters for PPP. |

### Disable

Use the **ppp disable** command to disable accounting for PPP.

**Syntax:**

**ppp disable** accounting

### List

Use the **ppp list** command to display the AAA configuration parameters for PPP.

**Syntax:**

**ppp list** all
accounting
authentication
authorization

config

## Set

Use the **ppp set** command to set the AAA configuration parameters for PPP.

**Syntax:**

**ppp set** aaa
accounting
authentication
authorization

**aaa** *authtype*
Sets the authentication, authorization, and accounting type. *Authtype* is one of the following:

**local** Sets the authentication, authorization, and accounting type to use a locally-maintained user database.

**remote**
Sets the authentication, authorization, and accounting type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**accounting** *authtype*
Sets the accounting type. *Authtype* is one of the following:

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authentication** *authtype*
Sets the authentication type. *Authtype* is one of the following:

**local** Sets the authentication type to use a locally-maintained user database.

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authorization** *authtype*
Sets the authorization type. *Authtype* is one of the following:

**local** Sets the authorization type to use a locally-maintained user database.

**remote**
Sets the authorization type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

# Servers

Use the **servers** command to configure individual remote AAA servers.

Table 124 lists the subcommands available with the **servers** command.

*Table 124. Server Subcommands*

| Command | Function |
|---|---|
| Add | Adds a remote AAA server profile. |
| Change | Changes a remote server profile. |
| Delete | Deletes a remote server profile. |
| Lists | Displays the AAA server profile information. |

## Add

Use the **servers add** command to add a remote server profile.

**Syntax:**

**servers add** name

**radius** Sets the authentication type to use the radius authentication server protocol.

Values for the following parameters can be set:

**key-for-encryption:**
Specifies the encryption key.

**Valid Values:** Any alphanumeric character string up to 32 characters long.

**Default Value:** None.

**primary-server-address:**
Specifies the address of the primary authentication server.

**Valid Values:** Any valid IP address

**Default Value:** 0.0.0.0

**retries**

**Valid Values:** 1 to 100

**Default Value:** 3

**retry-interval**

**Valid Values:** 1 to 60

**Default Value:** 3

**secondary-server-address:**
Specifies the address of the secondary authentication server.

**Valid Values:** Any valid IP address

**Default Value:** 0.0.0.0

**Author-Authent**
Specifies whether authorization attributes are transferred during authentication.

**Valid Values:** yes, no

**Default Value:** yes

**tacacs**

Sets the authentication type to use the TACACS authentication server protocol.

Values for the following parameters can be set:

**primary-server-address:**

Specifies the address of the primary authentication server.

**Valid Values:** Any valid IP address

**Default Value:** 0.0.0.0

**retries**

**Valid Values:** 1 to 100

**Default Value:** 3

**retry-interval**

**Valid Values:** 1 to 60

**Default Value:** 3

**secondary-server-address:**

Specifies the address of the secondary authentication server.

**Valid Values:** Any valid IP address

**Default Value:** 0.0.0.0

**tacacsplus**

Sets the authentication type to use the TACACS+ authentication server protocol.

Values for the following parameters can be set:

**encryption:**

Specifies whether encryption will be used.

**Valid Values:** yes, no

**Default Value:**

**key-for-encryption:**

Specifies the encryption key to be used.

**Valid Values:** Any 16-hexadecimal digit value

**Default Value:**

**primary-server-address:**

Specifies the address of the primary authentication server.

**Valid Values:** Any valid IP address

**Default Value:** 0.0.0.0

**privilege-level**

**Valid Values:** 0 through 15

**Default Value:** 0

**restarts**
Sets the number of restarts. This parameter does not include timeout restarts and only pertains to restarts requested by the server.

**Valid Values:** 0 to 3200

**Default Value:** 0

**time-to-connect**
The amount of time to allow to obtain the authentication from the server.

**Valid Values:** 1 to 60

**Default Value:** 9

**secondary-server-address:**
Specifies the address of the secondary authentication server.

**Valid Values:** Any valid IP address

**Default Value:** 0.0.0.0

## Change

Use the **servers change** command to change a remote server profile. See the **add** command for the remote server profile descriptions.

**Syntax:**

| | |
|---|---|
| **servers change** | radius |
| | tacacs |
| | tacacsplus |

See the **servers add** command for remote server profile descriptions.

## Delete

Use the **servers delete** command to delete a remote server profile. See the **add** command for the remote server profile descriptions.

**Syntax:**

| | |
|---|---|
| **servers delete** | radius |
| | tacacs |
| | tacacsplus |

See the **servers add** command for the remote server profile descriptions.

## List

Use the **servers list** command to display the AAA server profile information.

**Syntax:**

| | |
|---|---|
| **servers list** | all |
| | names |
| | profile |

# Set

Use the **set** command to set the parameters for login, PPP, and L2TP tunnel.

**Syntax:**

**set** aaa
accounting
authentication
authorization

**aaa** *authtype*
Sets the authentication, authorization, and accounting type. *Authtype* is one of the following:

**local** Sets the authentication, authorization, and accounting type to use a locally-maintained user database.

**remote**
Sets the authentication, authorization, and accounting type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**accounting** *authtype*
Sets the accounting type for login, PPP and tunnel. *Authtype* is one of the following:

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authentication** *authtype*
Sets the authentication type for login, PPP, tunnel. *Authtype* is one of the following:

**local** Sets the authentication type to use a locally-maintained user database.

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authorization** *authtype*
Sets the authorization type for login, PPP, and tunnel. *Authtype* is one of the following:

**local** Sets the authorization type to use a locally-maintained user database.

**remote**
Sets the authorization type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

# Tunnel

Use the **tunnel** command to configure AAA for L2TP tunnel.

Table 125 lists the subcommands available with the **tunnel** command.

*Table 125. Tunnel Subcommands*

| Command | Function |
|---|---|
| Disable | Disables accounting for L2TP tunnel. |
| List | Displays AAA configuration parameters for L2TP tunnel. |
| Set | Sets the AAA configuration parameters for L2TP tunnel. |

## Disable

Use the **tunnel disable** command to disable accounting for L2TP tunnel.

**Syntax:**

**tunnel disable** accounting

## List

Use the **tunnel list** command to display the AAA for L2TP tunnel.

**Syntax:**

**tunnel list** all
accounting
authentication
authorization
config

## Set

Use the **tunnel set** command to set the AAA configuration parameters for L2TP tunnel.

**Syntax:**

**tunnel set** aaa
accounting
authentication
authorization

**aaa** *authtype*
Sets the authentication, authorization, and accounting type. *Authtype* is one of the following:

**local** Sets the authentication, authorization, and accounting type to use a locally-maintained user database.

**remote**
Sets the authentication, authorization, and accounting type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**accounting** *authtype*
Sets the accounting type. *Authtype* is one of the following:

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authentication** *authtype*
Sets the authentication type. *Authtype* is one of the following:

**local** Sets the authentication type to use a locally-maintained user database.

**remote**
Sets the authentication type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

**authorization** *authtype*
Sets the authorization type. *Authtype* is one of the following:

**local** Sets the authorization type to use a locally-maintained user database.

**remote**
Sets the authorization type to use a remote user database.

**server id**
Specifies the identifier of the remote database.

# User-profiles

Use the **user-profiles** command to access the `User profile config>` command prompt. From this prompt, you can access the following commands.

*Table 126. User-profile Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add | Adds a PPP user profile. |
| Change | Changes a PPP user profile. |
| Delete | Deletes a PPP user profile. |
| Disable | Disables a PPP user profile. |
| Enable | Enables a PPP user profile. |
| List | Lists the PPP user profile information. |
| Report | Generates a PPP user profile report. |
| Reset-user | Resets a PPP user profile. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Add

Use the **add** command to add a user-profile.

**Syntax:**

**add** ppp-user

tunnel

```
User profile config> add ppp-user
Enter name:  []? ppp01
Password:
Enter again to verify:
Allow inbound access for user? (Yes, No): [Yes]
Will user be tunneled? (Yes, No): [No]
Number of days before password expiry[0-1000] [0]?
IP address: [0.0.0.0]?
Enable encryption for this user/port (y/n) [No]:
Disable user ? (Yes, No): [No]
     PPP user name: ppp01
            Expiry: <unlimited>
   User IP address: Interface Default
        Encryption: Not Enabled
            Status: Enabled
    Login Attempts: 0
    Login Failures: 0
  Lockout Attempts: 0
User 'ppp01' has been added
```

**Name** Enter the userid for the PPP user.

**Password**
Enter the password for the PPP user.

**Verify password**
Enter the password again exactly as before for verification.

**Allow inbound access**
Allows inbound access to this user profile.

**Valid values:** yes, no

**Default value:** no

**Will user be tunneled?**
Specifies whether the user is tunneled.

**Valid values:** yes, no

**Default value:** no

**Number of days**
The number of days before the password expires.

**Valid values:** 0 to 360

**Default value:** 180

**IP address**
The IP address

**Valid values:** any valid IP address

**Default value:**none

**Enable encryption**
Specifies whether encryption is to be enabled for this user/port.

**Valid values:** yes, no

**Default value:** no

**Disable user**
Allows you to disable a user-profile.

**Valid values:** yes, no

**Default value:** no

```
User profile config> add tunnel
Enter name:  []? tunnel01
Enter hostname to use when connecting to this peer: []? host01
set shared secret? (Yes, No): [No]
Tunnel-Server endpoint address: [0.0.0.0]?
       Tunnel name: tunnel01
          Endpoint: not configured
          Hostname: host01
User 'tunnel01' has been added
```

## Change

Use the **change** command to change a user-profile.

**Syntax:**

| | |
|---|---|
| **change** | ppp-user |
| | tunnel |

## Delete

Use the **delete** command to delete a user-profile.

**Syntax:**

| | |
|---|---|
| **delete** | ppp-user |
| | tunnel |

## Disable

Use the **disable** command to disable a user-profile.

**Syntax:**

| | |
|---|---|
| **disable** | *name* |

## Enable

Use the **enable** command to enable a user-profile.

**Syntax:**

| | |
|---|---|
| **enable** | *name* |

## List

Use the **list** command to list user-profile information.

**Syntax:**

| | |
|---|---|
| **list** | ppp-user |

tunnel

```
User profile config> list ppp-user
List  (Name, Verb, User, Addr, Encr, zdump): [Verb]
     PPP user name: ppp01
            Expiry: <unlimited>
   User IP address: Interface Default
        Encryption: Not Enabled
            Status: Enabled
    Login Attempts: 0
    Login Failures: 0
  Lockout Attempts: 0
1 record displayed.
```

**List** Specifies how to access the list information.

**Valid values:** name, verb, user, addr, encr, zdump

**Default value:** verb

**PPP user name**
Lists the user name.

**Expiry**
List the expiration date.

**User IP address**
List the users IP address.

**Encryption**
Lists whether encryption is enabled or not enabled.

**Status**
Lists whether status is enabled or not enabled

**Login attempts**
Lists the number of times the user has attempted to login.

**Login failures**
Lists the number of failed attempts to login.

**Lockout attempts**
Lists the number of lockout attempts.

## Report

Use the **report** command to generate a PPP user profile report.

**Syntax:**

**report**

addresses
all
callback
dialout
dump
encrypt
name
password
time
user

```
User profile config> report addresses
PPP user name      User IP address
-----------------  -----------------
ppp01              Interface Default
1 record displayed.

User profile config> report all
     PPP user name: ppp01
           Expiry: <unlimited>
   User IP address: Interface Default
        Encryption: Not Enabled
            Status: Enabled
    Login Attempts: 0
    Login Failures: 0
  Lockout Attempts: 0
1 record displayed.

User profile config> report callback
PPP user name      Callback type        Phone Number
-----------------  -------------------  --------------------
ppp01
1 record displayed.

User profile config> report dialout
PPP user name      Dial-out
-----------------  ---------------
ppp01
1 record displayed.

User profile config> report dump
Enter user name:  []? user01

User profile config> report encrypt
PPP user name      Encryption
-----------------  ---------------
ppp01              Not Enabled
1 record displayed.

User profile config> report name
PPP user name
-----------------
ppp01
1 record displayed.

User profile config> report password
PPP user name      Expiry        Grace
-----------------  ------------  ------
ppp01              <unlimited>
1 record displayed.

User profile config> report time
PPP user name      Time alotted
-----------------  -------------------
ppp01
1 record displayed.

User profile config> report user
Enter user name:  []? login01
     PPP user name: login01
        Expiry: <unlimited>
   User IP address: Interface Default
        Encryption: Not Enabled
```

## Reset-user

Use the **reset-user** command to reset a user-profile.

**Syntax:**

**reset-user** *name*

# Chapter 70. Overview of Encryption

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

The objective of encryption is to transform data into an unreadable form to ensure privacy. The ***encrypted*** data needs to be decrypted to get the original data.

Nways devices support Data Encryption Standard (DES) Cipher Block Chaining (CBC) mode. DES is a symmetric encryption standard that uses a 56-bit key for PPP or a 40-bit key for Frame Relay encryption and decryption.

You can encrypt data transmitted on either PPP or Frame Relay links. Encryption for PPP is described in RFC 1968 and 1969. Frame Relay encryption support is proprietary.

## PPP Encryption

The Encryption Control Protocol is used in the router to negotiate the use of encryption on the point-to-point links communicating using PPP protocol . The Encryption Control Protocol provides a generalized mechanism to negotiate which encryption and decryption algorithms will be used over a PPP link. Different encryption algorithms can be negotiated in each direction of the PPP link.

A method of encryption and decryption is called an ***encryption algorithm***. Encryption algorithms use a key to control encryption and decryption. Unlike compression, the router encrypts in both directions of the link, because encrypting in only one direction is a security risk. The link will be terminated whenever ECP cannot negotiate encryption algorithms in both directions.

## Configuring Encryption for PPP

To configure the device to use encryption at the data link layer, you should:

1. Set the encryption keys for remote devices and local PPP interfaces.

   Set the encryption key for the remote device using the **add ppp-user** command at the `Config>` prompt (see “Add” on page 68).

   Set the encryption key for the local PPP interface using the **set name** command (see “Set” on page 532).
2. Configure individual PPP links to use Encryption Control Protocol (ECP) by using the **enable ecp** command at the `PPP Config>` prompt (see “Enable” on page 527 ).
3. Enable PAP, CHAP, or SPAP.

You can also disable encryption, change the encryption key for a user, list the status of encryption, or set the name and encryption key the device uses when requesting encryption. For information about

- Disabling encryption, see the **disable ecp** command in “Disable” on page 526.
- Changing the user’s encryption key, see the **change ppp-user** command in “Change” on page 73.
- Listing the encryption status, see the **list ecp** command in “List” on page 528.

- Setting the device's name and encryption key, see the **set name** command in "Set" on page 532.

## Monitoring Encryption for PPP

You can monitor the various encryption settings on the interfaces by:

1. Accessing the monitoring prompt using the **talk 5** command.
2. Selecting the interface you want to monitor using the **network x** command. This command puts you at the `PPP x>` prompt.

From this prompt, you can:

- List the current state of encryption, the most recent encryption negotiation, the elapsed time since an encryption state change, and the algorithms in use by the encrypters. (See the **list control ecp** command on page 544.)
- List the encryption control packets received and transmitted on the interface. (See the **list ecp** command on page 555.)
- List the encrypted data packets transmitted or received on the interface. (see the list edp command on page 556.)

## Configuring Encryption on Frame Relay Interfaces

**Note:** Frame relay uses a proprietary encryption scheme.

Data encryption is supported on all interfaces on which you have enabled encryption. You can configure individual circuits on an encryption-enabled interface to perform or not perform encryption as desired.

To configure the device to use encryption on frame relay links:

1. Access the frame relay configuration prompt using the **talk 6** command.
2. Select the frame relay interface that you want to be encryption-capable using the **net** # command
3. Enable encryption on the frame relay interface using the **enable encryption** command. See "Enable" on page 483.
4. Add encryption—capable permanent virtual circuits and define the encryption key for each of the PVCs using the **add permanent-virtual-circuit** command. See "Add" on page 476.
5. Repeat steps 1 through 4 for each encryption-capable interface you are configuring.

**Note:** If encryption is enabled for a FR permanent virtual circuit then data will not flow over the circuit unless encryption is successfully negotiated with the device at the other end of the virtual circuit. Encryption is not supported for orphan circuits since you must configure the PVC in order to enter the encryption key.

You can also disable encryption for an interface, change the encryption settings for a PVC or list the status of encryption. For information about

- Disabling encryption on an interface, see the **disable encryption** command in "Disable" on page 481.

- Changing the encryption settings for a PVC, see the **change permanent-virtual-circuit** command in “Change” on page 479.
- Listing the encryption status, see the **list all,**, **list lmi,** and the **list permanent-virtual-circuit** commands in “List” on page 486.

## Monitoring Encryption on Frame Relay Interfaces

You can monitor the various encryption settings on the interfaces by:

1. Accessing the monitoring prompt using the **talk 5** command.
2. Selecting the interface you want to monitor using the **network** *#* command. This command puts you at the FR *x*> prompt.

From this prompt, you can list the current encryption state for an interface, a PVC, or a circuit. See “List” on page 499.

# Chapter 71. Using Quality of Service (QoS)

This chapter describes how to use the Quality of Service (QoS) feature in the device.

## Quality of Service Overview

The QoS feature leverages the benefits of ATM QoS capabilities for LAN Emulation Data Direct VCCs. This support is referred to as “Configurable QoS for LAN Emulation”. The key attributes and the benefits of this feature are as follows:

- An LE Client makes use of configured QoS parameters for its Data Direct VCCs.
- QoS parameters can be configured for:
  - LE Client
  - ATM Interface
- The set of QoS parameters configured are for use with ATM Forum UNI 3.0/3.1 signaling. The parameters include the desired Peak Cell Rate, Sustained Cell Rate, QoS Class and Maximum Burst Size.
- Maximum Reserved Bandwidth per VCC can be configured to protect an LE Client from accepting/establishing VCCs whose traffic parameters it cannot support.
- The QoS Negotiation mechanism enables the participating LE Clients to be aware of each other’s QoS parameters. A data-direct VCC is set up using the negotiated parameters.

## Benefits of QoS

- Using QoS for the LE Client, ATM Interface, or Emulated LAN provides the following benefits for LANE Data Direct VCCs.
  - An LE Client can be configured with QoS if the QoS required by the client is different from the QoS required by other clients on the ELAN. For example, if an LE Client serves a file server, then the user may want to configure appropriate QoS parameters for all traffic to and from the file server.
  - An Emulated LAN can be configured with QoS if the user wishes to provide QoS for all traffic in that ELAN. For example, an ELAN carrying SNA traffic can be given priority by configuring QoS parameters for that ELAN.
  - An ATM Interface can be configured with QoS if a user wants all LE Clients on that ATM interface to use the same set of parameters. For example, if an ATM Interface is connected at 25 Mbps, the user can configure appropriate parameters that are different from those at a 155-Mbps interface.

# Chapter 72. Configuring and Monitoring Quality of Service (QoS)

This chapter describes Quality of Service (QoS) configuration and operational commands for LAN and ELAN interfaces in the router. It contains the following sections:



## QoS Configuration Parameters

This section describes nine parameters that are used for QoS configuration. The following six parameters can be configured for an LE Client, ATM Interface, and an Emulated LAN:

1. max-reserved-bandwidth
2. traffic-type
3. peak-cell-rate
4. sustained-cell-rate
5. max-burst-size
6. qos-class

The following two parameters can be configured for an Emulated LAN and an LE Client:

1. *validate-pcr-of-best-effort-vccs*
2. *negotiate-qos*

The *accept-qos-parms-from-lecs* parameter can be configured only for an LE Client.

The first six parameters control the traffic characteristics of Data Direct VCCs established by the LE Client while the first parameter also applies to the calls received by the LE Client. The following characteristics are associated with all the Data Direct VCCs established by the LE Client:

- Bandwidth is not reserved for best-effort traffic.
- Traffic parameters apply to both forward and backward directions.
- When a reserved bandwidth connection is rejected due to the traffic parameters or QoS Class, the call is retried as a best-effort connection with the configured peak cell rate (cause codes on release or release-complete messages are used to determine why a VCC was released).

- When a best-effort connection is rejected due to the Peak Cell Rate (PCR), the call may be automatically retried with a lower PCR. Retries are performed under the following conditions:
  1. If the rejected PCR is greater than 100 Mbps, the call is retried with a PCR of 100 Mbps.
  2. Otherwise, if the rejected PCR is greater than 25 Mbps, the call is retried with a PCR of 25 Mbps.

## Maximum Reserved Bandwidth (max-reserved-bandwidth)

The maximum reserved bandwidth acceptable for a Data Direct VCC. This parameter applies to both Data Direct VCC calls received by the LE Client and Data Direct VCC calls placed by the LE Client. For incoming calls, this parameter defines the maximum acceptable SCR for a Data Direct VCC. If SCR is not specified on the incoming call, then this parameter defines the maximum acceptable PCR for a Data Direct VCC with reserved bandwidth.

Calls received with traffic parameters specifying higher rates will be released. If SCR is specified on the incoming call, the call will not be rejected due to the PCR or Maximum Burst Size. The constraint imposed by this parameter is not applicable to BEST_EFFORT connections. For outgoing calls, this parameter sets an upper bound on the amount of reserved bandwidth that can be requested for a Data Direct VCC. Therefore the traffic-type and sustained-cell-rate parameters are dependent upon this parameter.

**Valid Values:**
Integer in the range 0 to the line speed of ATM device in Kbps

**Default Value:**
0

## Traffic Type (traffic-type)

The desired traffic type for Data Direct VCCs. If QoS parameters are not negotiated, then this parameter specifies the type of calls placed by the LE Client. Otherwise, if QoS parameters are negotiated, this parameter specifies the desired type of traffic characteristics for Data Direct VCCs. When QoS parameters are negotiated, if either the source or target LEC desires a reserved bandwidth connection and both LECs support reserved bandwidth connections (that is, max-reserved-bandwidth > 0), then an attempt will be made to establish a reserved bandwidth Data Direct VCC between the two LECs. Otherwise, the Data Direct VCC will be a best-effort connection. Dependencies: max-reserved-bandwidth

**Valid Values:**
best_effort or reserved_bandwidth

**Default:**
best_effort

## Peak Cell Rate (peak-cell-rate)

The desired peak cell rate for Data Direct VCCs. If QoS parameters are not negotiated, then this parameter specifies the PCR traffic parameter for Data Direct VCC calls placed by the LE Client. Otherwise, if QoS parameters are negotiated,

this parameter specifies the desired PCR traffic parameter for Data Direct VCCs. The minimum of the desired PCRs of the two LECs is used for negotiated best-effort VCCs.

When a reserved bandwidth VCC is negotiated and only one of the LE Clients requests a reserved bandwidth connection, then the desired PCR of that LEC is used for the Data Direct VCC subject to the upper bound imposed by the line rate of the local ATM device. If both LECs request a reserved bandwidth connection, then the maximum of the desired PCRs of the LE Clients is used for the Data Direct VCC subject to the upper bound imposed the line rate of the local ATM device.

**Valid Values:**

An integer value in the range 0 to the line speed of ATM device in Kbps

**Default Value:**

Line speed of LEC ATM Device in Kbps.

## Sustained Cell Rate (sustained-cell-rate)

The desired sustained cell rate for Data Direct VCCs. If QoS parameters are not negotiated, then this parameter specifies the SCR traffic parameter for Data Direct VCC calls placed by the LE Client. Otherwise, if QoS parameters are negotiated, this parameter specifies the desired SCR traffic parameter for Data Direct VCCs.

When a reserved bandwidth VCC is negotiated and only one of the LE Clients requests a reserved bandwidth connection, then the desired SCR of that LEC is used for the Data Direct VCC (subject to the upper bound imposed by the max-reserved-bandwidth parameter of the other LEC). If both LECs request a reserved bandwidth connection, then the maximum of the desired SCRs of the LE Clients is used for the Data Direct VCC (subject to the upper bound imposed by the max-reserved-bandwidth parameters of both LECs). In any case (negotiation or not), if the SCR that is to be signaled equals the PCR that is to be signaled, then the call is signaled with PCR only.

Dependencies: max-reserved-bandwidth, traffic-type and peak-cell-rate. This parameter is applicable only when traffic-type is RESERVED_BANDWIDTH.

**Valid Values:**

An integer value in the range 0 to the minimum of max-reserved-bandwidth and peak-cell-rate, specified in Kbps

**Default Value**

None

## Maximum Burst Size (max-burst-size)

The desired maximum burst size for Data Direct VCCs. If QoS parameters are not negotiated, then this parameter specifies the Maximum Burst Size traffic parameter for Data Direct VCC calls placed by the LE Client. Otherwise, if QoS parameters are negotiated, this parameter specifies the desired Maximum Burst Size traffic parameter for Data Direct VCCs.

When a reserved bandwidth VCC is negotiated and only one of the LE Clients requests a reserved bandwidth connection, then the desired Maximum Burst Size of that LEC is used for the Data Direct VCC. If both LECs request a reserved bandwidth connection, then the maximum of the desired Maximum Burst Sizes of the LE Clients is used for the Data Direct VCC.

In any case (negotiation or not), the Maximum Burst Size is signaled only when SCR is signaled. Although this parameter is expressed in units of cells, it is configured as an integer multiple of the Maximum Data Frame Size (specified in LEC's C3 parameter) with a lower bound of 1.

Dependencies: This parameter is applicable only when traffic-type is RESERVED_BANDWIDTH.

**Valid Values:**
An integer number of frames; must be greater than 0

**Default:**
1 frame

## QoS Class (qos-class)

The desired QoS class for reserved bandwidth calls. If QoS parameters are not negotiated, then this parameter specifies the QoS Class to be used for reserved bandwidth Data Direct VCC calls placed by the LE Client. Otherwise, if QoS parameters are negotiated, this parameter specifies the QoS Class that is desired for Data Direct VCCs. Unspecified QoS Class is always used on best-effort calls. Specified QoS Classes define objective values for ATM performance. Specified QoS Classes define objective values for ATM performance parameters such as cell loss ratio and cell transfer delay.

The UNI Specification states that:

**Specified QoS Class 1**
should yield performance comparable to current digital private line performance.

**Specified QoS Class 2**
is intended for packetized video and audio in teleconferencing and multimedia applications.

**Specified QoS Class 3**
is intended for interoperation of connection oriented protocols, such as frame relay.

**Specified QoS Class 4**
is intended for interoperation of connectionless protocols, such as IP or SMDS.

LECs must be able to accept calls with any of the above QoS Classes. When QoS parameters are negotiated, the configured QoS Classes of the two LECs are compared, and the QoS Class with the more stringent requirements is used.

**Valid Values:**
0: for Unspecified QoS Class

1: for Specified QoS Class 1

2: for Specified QoS Class 2

3: for Specified QoS Class 3

4: for Specified QoS Class 4

**Default Value:**
0 (Unspecified QoS Class)

## Validate PCR of Best-Effort VCCs (validate-pcr-of-best-effort-vccs)

To validate Peak Cell Rate of Best-Effort VCCs. When FALSE, best-effort VCCs will be accepted without regard to the signaled forward PCR. When TRUE, best- effort VCCs will be rejected if the signaled forward PCR exceeds the line rate of the LE Client ATM device. Calls will not be rejected due to the backward PCR. The signaled backward PCR will be honored if it does not exceed the line rate; otherwise, transmissions to the caller will be at line rate.

**Notes:**

1. Accepting best-effort VCCs with forward PCRs that exceed the line rate can result in poor performance due to excessive retransmissions; however, rejecting these VCCs can result in interoperability problems.
2. The YES setting is useful when callers will retry with a lower PCR following call rejection due to unavailable cell rate.

**Valid Values:**
yes, no

**Default Value:**
no

## Negotiate QoS (negotiate-qos)

Enable QoS parameter negotiation for Data Direct VCCs. This parameter should be enabled only when connecting to an IBM MSS LES. When this parameter is YES, the LE Client will include an IBM Traffic Parameter TLV in LE_JOIN_REQUEST and LE_ARP_RESPONSE frames sent to the LES. This TLV will include the values of max-reserved-bandwidth, traffic-type, peak-cell-rate, sustained-cell-rate, max-burst-size and qos-class. An IBM Traffic Parameter TLV may also be included in a LE_ARP_RESPONSE returned to the LE Client by the LES.

If there is no TLV in a LE_ARP_RESPONSE received by the LE Client, then the local configuration parameters must be used to setup the Data Direct VCC. If a TLV is included in a LE_ARP_RESPONSE, the LE Client must compare the contents of the TLV with the corresponding local values to determine the “negotiated” or “best” set of parameters acceptable to both parties before signalling for the Data Direct VCC.

**Valid Values:**
yes, no

**Default Value:**
no

## Accept QoS Parms from LECS (accept-qos-parms-from-lecs)

This parameter gives the ability to configure an LE Client to accept/reject QoS parameters from a LECS. When this parameter is YES, the LE Client should use the QoS parameters obtained from the LE Clients in the LE_CONFIGURE_RESPONSE frames, that is, the QoS parameters from the LE Clients override the locally configured QoS parameters. If this parameter is NO then the LE Client will ignore any QoS parameters received in an LE_CONFIGURE_RESPONSE frame from the LE Clients.

**Valid Values:**
yes, no

**Default Value:**
no

## Accessing the QoS Configuration Prompt

Use the **feature** command from the CONFIG process to access the Quality of Service configuration commands. Enter **feature** followed by the feature number (6) or short name (QOS). For example:

```
Config> feature qos
Quality of Service - Configuration
QoS Config>
```

Once you access the QoS Config> prompt, you can configure the Quality of Service (QoS) of an LE Client, or an ATM Interface. To return to the Config> prompt at any time, enter the **exit** command at the QoS Config> prompt.

Alternatively, you can configure QoS parameters for an LE Client or an ATM Interface by accessing the entities as follows:

- LE Client
  1. At the Config> prompt, enter the **network** command and the LE Client interface number.
  2. At the LE Client configuration> prompt enter **qos-configuration**.

     **Example:**

     ```
     config> network 3
     Token Ring Forum Compliant LEC Config> qos-configuration
     LEC QoS Config>
     ```
- ATM Interface
  1. at the Config> prompt, enter the **network** command and the ATM interface number to get you to the ATM Config> prompt.
  2. Enter the **interface** parameter to get to the ATM Interface Config> prompt.
  3. At the ATM InterfaceConfig> prompt enter **qos-configuration**.

     **Example:**

     ```
     config> network 0
     ATM Config> interface
     ATM Interface Config> qos-configuration
     ATM-I/F 0 QoS>
     ```

## Quality of Service Commands

This section summarizes the QoS configuration commands. Use the following commands to configure Quality of Service. Enter the commands from the QoS Config> prompt.

*Table 127. Quality of Service (QoS) Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| le-client | Gets you to the LE Client QoS configuration > prompt for the selected LE client. |
| atm-interface | Gets you to the ATM Interface QoS configuration> prompt for the selected ATM interface. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

# LE Client QoS Configuration Commands

This section summarizes and explains the commands for configuring QoS for a specific LE Client.

Use the following commands at the `LEC QoS config>` prompt.

*Table 128. LE Client Quality of Service (QoS) Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists the current QoS configuration of the LE Client. |
| Set | Sets the QoS parameters of the LE Client. |
| Remove | Removes the QoS configuration of the LE Client. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to list the QoS configuration of this LE Client. QoS parameters are listed only if at least one has been specifically configured (see Example 1). Otherwise, no parameters are listed (see Example 2).

**Syntax:**

**list**

**Example 1:**

```
LEC QoS Config> list

           LE Client QoS Configuration for Data Direct VCCs
           ========================================
     (ATM interface number = 0,  LEC interface number = 3)

  Maximum Reserved Bandwidth for a Data-Direct VCC = 10000 Kbps
  Data-Direct VCC Type ........................... = Best-Effort
  Data-Direct VCC Peak Cell Rate ................. = 155000 Kbps
  Data-Direct VCC Sustained Cell Rate ............ = 155000 Kbps
  Desired QoS Class of Reserved Connections ...... = 0
  Max Burst Size of Reserved Connections    ...... = 0 frames

  Validate Peak Rate of Best-Effort connections .. = No
  Enable QoS Parameter Negotiation ............... = Yes
  Accept QoS Parameters from LECS ................ = Yes

LEC QoS Config>
```

**Example 2:**

```
LEC QoS Config> list

   QoS has not been configured for this LEC.
   Please use the SET option to configure QoS.

LEC QoS Config>
```

## Set

Use the **set** command to specify LE Client QoS parameters.

**Syntax:**

**set**

accept-qos-parms-from-lecs

all-default-values

max-burst-size

max-reserved-bandwidth

negotiate-qos

peak-cell-rate

qos-class

sustained-cell-rate

traffic-type

validate-pcr-of-best-effort-vccs

**accept-qos-parms-from-lecs**

Use this option to enable/disable the LE Client to accept/reject the QoS parameters received from an LECS as TLVs. See “Accept QoS Parms from LECS (accept-qos-parms-from-lecs)” on page 853 for a more detailed description of this parameter.

**Valid Values:**

YES, NO

**Default Value:**

YES

**Example:**

```
LEC QoS Config> se acc y
LEC QoS Config>
```

**all-default-values**

Use this option to set the QoS parameters to default values. In the following example the default values are also listed.

**Example:**

```
LEC QoS Config> set all-default-values
   Failed to locate existing QoS configuration record!
   Using a new set of default values ...
   Initializing all parameters to default values
   LEC QoS Config>  list

              LE Client QoS Configuration for Data Direct VCCs
              ========================================
         (ATM interface number = 0,  LEC interface number = 3)

     Maximum Reserved Bandwidth for a Data-Direct VCC = 0 Kbps
     Data-Direct VCC Type ........................... = Best-Effort
     Data-Direct VCC Peak Cell Rate ................. = 155000 Kbps
     Data-Direct VCC Sustained Cell Rate ............ = 155000 Kbps
     Desired QoS Class of Reserved Connections ...... = 0
     Max Burst Size of Reserved Connections    ...... = 0 frames

     Validate Peak Rate of Best-Effort connections .. = No
     Enable QoS Parameter Negotiation ............... = No
     Accept QoS Parameters from LECS ................ = Yes

  LEC QoS Config>
```

**max-burst-size**

Sets the desired maximum burst size in frames. See “Maximum Burst Size (max-burst-size)” on page 851 for a more detailed description of this parameter.

**Valid Values:**

An integer number of frames; must be greater than 0

**Default:**
1 frame

**Example:**

```
LEC QoS Config> se ma
   Maximum Burst Size in Kbps [1]? 10000
   LEC QoS Config>
```

**max-reserved-bandwidth**
Use this option to set the maximum reserved bandwidth allowable per Data Direct VCC. See “Maximum Reserved Bandwidth (max-reserved-bandwidth)” on page 850 for a more detailed description of this parameter.

**Valid Values:**
Integer in the range 0 to the line speed of ATM device in Kbps

**Default Value:**
0

**Example:**

```
   LEC QoS Config> set max-reserved-bandwidth
   Maximum reserved bandwidth acceptable for a data-direct VCC (in Kbps) [0]? 20000
   LEC QoS Config>
```

**negotiate-qos**
Use this option to enable/disable the LE Client’s participation in QoS negotiation. See “Negotiate QoS (negotiate-qos)” on page 853 for a more detailed description of this parameter.

**Valid Values:**
YES, NO

**Default Value:**
NO

**Example:**

```
LEC QoS Config> se neg y
LEC QoS Config>
```

**peak-cell-rate**
Sets the desired peak cell rate for Data Direct. See “Peak Cell Rate (peak-cell-rate)” on page 850 for a more detailed description of this parameter.

**Valid Values:**
An integer value in the range 0 to the line speed of ATM device in Kbps

**Default Value:**
Line speed of LEC ATM Device in Kbps.

**Example:**

```
   LEC QoS Config> set peak-cell-rate
   Data-Direct VCC Peak Cell Rate in Kbps [1]? 25000
   LEC QoS Config>
```

**qos-class**
Sets the desired QoS Class for Data Direct VCCs. See “QoS Class (qos-class)” on page 852 for a more detailed description of this parameter.

**Valid Values:**
0: for Unspecified QoS Class

1: for Specified QoS Class 1

2: for Specified QoS Class 2

3: for Specified QoS Class 3

4: for Specified QoS Class 4

**Default Value:**
0 (Unspecified QoS Class)

**Example:**

```
LEC QoS Config> se qos
Desired QoS Class for Data Direct VCCs [0]? 1
LEC QoS Config>
```

**sustained-cell-rate**
Sets the desired sustained cell rate for Data Direct VCCs. See “Sustained Cell Rate (sustained-cell-rate)” on page 851 for a more detailed description of this parameter.

**Valid Values:**
An integer value in the range 0 to the minimum of max-reserved-bandwidth and peak-cell-rate, specified in Kbps

**Default Value**
None

**Example:**

```
LEC QoS Config> se sus
Data-Direct VCC Sustained Cell Rate in Kbps [1]? 10000
LEC QoS Config>
```

**traffic-type**
Sets the desired traffic for Data Direct VCCs. See “Traffic Type (traffic-type)” on page 850 for a more detailed description of this parameter.

**Valid Values:**
BEST_EFFORT or RESERVED_BANDWIDTH

**Default:**
BEST EFFORT.

**Example:**

```
LEC QoS Config>set traffic-type
  Choose from:
     (0): Best-Effort
     (1): Reserved-Bandwidth
Data Direct VCC Type [0]? 1
NOTE: Peak Cell Rate has been reset to 1
      Sustained Cell Rate has been reset to 1
      Max Reserved Bandwidth has been reset to 1
      Please configure appropriate values.
LEC QoS Config>
```

**validate-pcr-of-best-effort-vccs**
Use this option to enable/disable validation of the Peak Cell Rate traffic parameter of the Data Direct VCC calls received by this LE Client. See “Validate PCR of Best-Effort VCCs (validate-pcr-of-best-effort-vccs)” on page 853 for a more detailed description of this parameter.

**Valid Values:**
YES, NO

**Default Value:**
NO

**Example:**

```
LEC QoS Config> se val y
LEC QoS Config>
```

## Remove

Use the **remove** command to remove the QoS configuration of this LE Client.

**Syntax:**

**remove**

**Example:**

```
LEC QoS Config> remove
WARNING: This option deletes the QoS configuration.
         To re-configure use any of the SET options.
Should the LEC QoS configuration be deleted? [No]: yes
Deleted QoS configuration successfully
LEC QoS Config>
```

# ATM Interface QoS Configuration Commands

*Table 129. LE Client Quality of Service (QoS) Configuration Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists the current ATM Interface QoS configuration. |
| Set | Sets the ATM Interface QoS parameters. |
| Remove | Removes the QoS configuration of the ATM Interface. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to list the QoS configuration of this ATM Interface. QoS parameters are listed only if at least one parameter has been configured (see following example). Otherwise, no parameters are listed.

**Syntax:**

**list**

**Example:**

```
ATM-I/F 0 QoS> list

          ATM Interface 'Quality of Service' Configuration
          ================================================
                    (ATM interface number = 0  )

     Maximum Reserved Bandwidth for a VCC = 15000 Kbps
     VCC Type ........................... = RESERVED-BANDWIDTH
     Peak Cell Rate ..................... = 20000 Kbps
     Sustained Cell Rate ................ = 5000 Kbps
     QoS Class .......................... = 4
     Maximum Burst Size ................. = 5 frames
ATM-I/F 0 QoS>
```

# Set

Use the **set** command to specify ATM Interface QoS parameters.

**Syntax:**

**set** max-burst-size
max-reserved-bandwidth
peak-cell-rate
qos-class
sustained-cell-rate
traffic-type

**max-burst-size**

Sets the desired maximum burst size in frames. See “Maximum Burst Size (max-burst-size)” on page 851 for a more detailed description of this parameter.

**Valid Values:**

An integer number of frames; must be greater than 0

**Default:**

1 frame

**Example:**

```
ATM-I/F 0 QoS Config> se ma
Maximum Burst Size in Kbps [1]? 10000
ATM-I/F 0 QoS Config>
```

**max-reserved-bandwidth**

Use this option to set the maximum reserved bandwidth allowable for each Data Direct VCC. See “Maximum Reserved Bandwidth (max-reserved-bandwidth)” on page 850 for a more detailed description of this parameter.

**Valid Values:**

Integer in the range 0 to the line speed of ATM device in Kbps

**Default Value:**

0

**Example:**

```
ATM-I/F 0 QoS> se max-reserved-bandwidth
Maximum reserved bandwidth acceptable for a data-direct VCC (in Kbps) [0]?
15000
ATM-I/F 0 QoS>
```

**peak-cell-rate**

Sets the desired peak cell rate for Data Direct VCCs. See “Peak Cell Rate (peak-cell-rate)” on page 850 for a more detailed description of this parameter.

**Valid Values:**

An integer value in the range 0 to the line speed of ATM device in Kbps

**Default Value:**

Line speed of LEC ATM Device in Kbps.

**Example:**

```
ATM-I/F 0 QoS Config> set peak-cell-rate
Data-Direct VCC Peak Cell Rate in Kbps [1]? 25000
ATM-I/F 0 QoS Config>
```

**qos-class**

Sets the desired QoS Class for Data Direct VCCs. See “QoS Class (qos-class)” on page 852 for a more detailed description of this parameter.

**Valid Values:**

0: for Unspecified QoS Class

1: for Specified QoS Class 1

2: for Specified QoS Class 2

3: for Specified QoS Class 3

4: for Specified QoS Class 4

**Default Value:**

0 (Unspecified QoS Class)

**Example:**

```
ATM-I/F 0 QoS Config> se qos
Desired QoS Class for Data Direct VCCs [0]? 1
ATM-I/F 0 QoS Config>
```

**sustained-cell-rate**

Sets the desired sustained cell rate for Data Direct VCCs. See “Sustained Cell Rate (sustained-cell-rate)” on page 851 for a more detailed description of this parameter.

**Valid Values:**

An integer value in the range 0 to the minimum of max-reserved-bandwidth and peak-cell-rate; specified in Kbps

**Default Value**

None

**Example:**

```
ATM-I/F 0 QoS Config> se sus
Data-Direct VCC Sustained Cell Rate in Kbps [1]? 10000
ATM-I/F 0 QoS Config>
```

**traffic-type**

Sets the desired traffic for Data Direct VCCs. See “Traffic Type (traffic-type)” on page 850 for a more detailed description of this parameter.

**Valid Values:**

BEST_EFFORT or RESERVED_BANDWIDTH

**Default:**

BEST EFFORT.

**Example:**

```
ATM-I/F 0 QoS> set traffic-type
  Choose from:
     (0): Best-Effort
     (1): Reserved Bandwidth
Traffic Type of VCCs [1]? 0
ATM-I/F 0 QoS>
```

## Remove

Use the **remove** command to remove the QoS configuration of this ATM Interface.

**Syntax:**

**remove**

**Example:**

```
ATM-I/F 0 QoS> remove
WARNING: This option deletes the QoS configuration.
         To re-configure use any of the SET options.
Should the ATM Interface QoS configuration be deleted? [No]: yes
Deleted QoS SRAM record successfully
ATM-I/F 0 QoS>
```

# Accessing the QoS Monitoring Commands

Use the **feature** command from the GWCON process to access the Quality of Service monitoring commands. Enter the **feature** followed by the feature number (6) or short name (QOS). For example:

```
+feature qos
Quality of Service (QoS) - User Monitoring
QoS+
```

Once you access the QoS monitoring prompt, you can select the monitoring of a particular LE Client. To return to the GWCON prompt at any time, enter the exit command at the QoS monitoring prompt.

Alternatively, you can access the QoS Monitoring of an LE Client as follows:

1. At the GWCON prompt (+), enter the network command and the LE Client interface number.
2. At the LE Client monitoring prompt enter **qos-information**.

   **Example:**

```
+network 3
ATM Emulated LAN Monitoring
LEC+qos information
LE Client QoS Monitoring
LEC 3 QoS+
```

# Quality of Service Monitoring Commands

This section summarizes the QoS monitoring commands. Enter these commands at the QoS+ prompt.

*Table 130. Quality of Service (QoS) Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| le-client | Gets you to the `LE Client QoS console +` prompt for the selected LE client. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

# LE Client QoS Monitoring Commands

This section summarizes the LE Client QoS monitoring commands. Enter the commands from the `LEC num QoS+` prompt.

*Table 131. LE Client QoS Monitoring Command Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists the current LE Client QoS information. Options include: configuration parameters, TLVs, VCCs, and statistics. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to list the QoS related information of this LE Client.

**Syntax:**

**list** configuration-parameters

data-direct-VCCs (Detailed Information)

statistics

tlv-information

vcc-information

**configuration-parameters**

Lists the QoS configuration parameters. Because parameters can be configured for an LE Client, ATM Interface or the ELAN, these parameters are displayed along with a resolved set of parameters that are used by the LE Client.

**le-client**

The parameters configured for this LE Client which are obtained from the SRAM records. If the SRAM records contain an invalid set of parameters then this column will not display any parameters values.

**ATM Interface**

The parameters configured for the ATM Interface used by this LE Client. These parameters are obtained from the local SRAM records. If the SRAM records contain an invalid set of parameters then this column will not display any parameter values.

**From LECS**

The parameters received by this LE Client from the LE Configuration Server. The parameters are received as individual TLVs in the LE_CONFIGURE_RESPONSE control message.

**used** The resolved set of traffic parameters which are used by for its Data Direct VCCs. If none of the entities is configured with QoS parameters, then the USED parameters represent the default parameters. If parameters are configured for at least one entity, then they are resolved as follows:

- If only the LE Client or the ATM Interface is configured with parameters and either the accept-parms-from-lecs is FALSE or no parameters were received from the LECS, then the configured LE Client or the ATM Interface parameters are used.
- If both the LE Client and the ATM Interface have configured parameters, then the LE Client parameters are used.
- If the accept-parms-from-lecs is TRUE and parameters were received from the LECS, then the LE Client parameters (or the default if the LE Client is not configured) are combined with those received from the LECS to form a complete set of the first six QoS parameters described in “QoS Configuration Parameters” on page 849.
- If the set of the first six QoS parameters described in “QoS Configuration Parameters” on page 849 contains an invalid combination then the parameters from the LECS are rejected. Note that the two flags negotiate-qos and validate-pcr-of-best-effort-vccs are validated independently.

**Example:**

```
LEC 1 QoS+ list configuration parameters


                ATM LEC Configured QoS  Parameters
                ==================================

      QoS                                        |    LEC      ATM-IF    FROM

   PARAMETER                              USED    |    SRAM     SRAM      LECS

 ------------------------------------------------------------------------------
 Max Reserved Bandwidth (cells/sec) :    23584   |    23584        0     none
                       (Kbits/sec) :    10000   |    10000        0     none
 VCC Type ......................... :   ResvBW   |   ResvBW   BstEft        0
 Peak Cell Rate ........(cells/sec) :    18867   |    18867   365566   365566
                       (Kbits/sec) :     8000   |     8000   155000   155000
 Sustained Cell Rate ...(cells/sec) :    18867   |    18867   365566     none
                       (Kbits/sec) :     8000   |     8000   155000     none
 QoS Class ........................ :        4   |        4        0     none
 Max Burst Size .............(cells) :       95   |       95        0     none
                           (frames) :        1   |        1        0     none
 Validate PCR of Best-Effort VCCs . :       NO   |       NO      n/a     none
 Enable QoS Negotiation ........... :      YES   |      YES      n/a     none
 Accept QoS Parameters from LECS .. :      YES   |      YES      n/a      n/a
 ------------------------------------------------------------------------------
(BstEft = Best Effort, ResvBW = Reserved Bandwidth)
(n/a = not applicable, none = no value is specified)

LEC 1 QoS+
```

### data-direct-vccs (Detailed Information)

This option lists the Data Direct VCC information of this LE Client. Similar information is also listed using **list vcc-information**.

**Example:**

```
LEC 1 QoS+ list data direct vccs

          LEC Data Direct VCCs - QoS Information
          ======================================

Conn Handle = 80, VPI = 0, VCI = 546
        Connection Type = RETRIED CONNECTION PARAMETERS
          TrafficType    = BEST EFFORT VCC
          PCR            = 58962 (25 Mbps)
          SCR            = 58962 (25 Mbps)
          QoS Class      = 0
          Max Burst Size = 0

Conn Handle = 78, VPI = 0, VCI = 544
        Connection Type = PARAMETERS SET BY DESTINATION
          TrafficType    = RESERVED BANDWIDTH VCC
          PCR            = 58962 (25 Mbps)
          SCR            = 16509 (7 Mbps)
```

```
          QoS Class      = 1
          Max Burst Size = 95

LEC 1 QoS+
```

**statistics**

Counters are maintained for the following statistics:

**Successful QoS Connections**
Number of RESERVED-BANDWIDTH connections established by the LE Client.

**Successful Best-Effort Connections**
Number of BEST-EFORT connections established by the LE Client.

**Failed QoS Connections**
Number of RESERVED-BANDWIDTH connection requests made by the LE Client that failed.

**Failed Best-Effort Connections**
Number of BEST-EFFORT connection requests made by the LE Client that failed.

**QoS Negotiation Applied**
Number of times the QoS negotiation extension was applied. Parameters are negotiated if the LE Client receives the destination LE Client's parameters in an LE_ARP_RESPONSE control message.

**PCR Proposal (IBM) Applied**
Number of times the IBM Peak Cell Rate Proposal was applied. This proposal recommends using specific rate parameters if signaling at 100 Mbps or 155 Mbps for BEST-EFFORT connections. This allows other participating IBM products (for example, 25-Mbps ATM adapters) to reject a connection based on the signaled peak cell rates.

**QoS Connections Accepted**
Number of RESERVED-BANDWIDTH connections accepted by this LE Client.

**Best-Effort Connections Accepted**
Number of BEST-EFFORT connections accepted by this LE Client.

**QoS Connections Rejected**
Number of RESERVED-BANDWIDTH connection requests received by this LE Client that were rejected.

**Best-Effort Connections Rejected**
Number of BEST-EFFORT connection requests received by this LE Client that were rejected.

**Rejected due to PCR Validation**
Number of BEST-EFFORT connections rejected by the LE Client due to validation of Peak Cell Rate when the validate-pcr-of-best-effort-vccs parameter is TRUE.

**Example:**

```
LEC 1 QoS+ li stat

QoS Statistics: of Data Direct Calls Placed by the LEC
------------------------------------------------------
    Successful QoS Connections          = 0
    Successful Best-Effort Connections  = 1
    Failed QoS Connections              = 1
    Failed Best-Effort Connections      = 1
```

```
    Qos Negotiation Applied             = 0
    PCR Proposal (IBM) Applied          = 0

QoS Statistics: of Data Direct Calls Received by the LEC
---------------------------------------------------------
    QoS Connections Accepted            = 1
    Best-Effort Connections Accepted    = 0
    QoS Connections Rejected            = 0
    Best-Effort Connections Rejected    = 0
    Rejected due to PCR Validation      = 0

LEC 1 QoS+
```

### tlv-information

Lists the IBM Traffic Information TLV that this LE Client registered with the LE Server. The TLV is registered only if the LE Client is participating in QoS Negotiation.

#### Example:

```
LEC 1 QoS+ list tlv

  Traffic Info TLV of the LEC (registered with the LES)
  =====================================================
    TLV Type .........................= 268458498
    TLV Length .......................= 24
    TLV Value:
           Maximum Reserved Bandwidth = 23584 cells/sec (10 Mbps)
           Data Direct VCC Type...... = RESERVED BANDWIDTH VCC
           Data Direct VCC PCR....... = 18867 cells/sec (8 Mbps)
           Data Direct VCC SCR....... = 18867 cells/sec (8 Mbps)
           Data Direct VCC QoS Class  = 4
           Maximum Burst Size         = 95 cells (1 frames)

LEC 1 QoS+
```

### vcc-information

Lists all active VCCs of the LE Client. The information includes the traffic parameters of the connections. For BEST-EFFORT connections, the Sustained Cell Rate is displayed to be the same as the Peak Cell Rate, QoS Class and the Maximum Burst Size are displayed as 0.

The Parameter Descriptor entries are:

**SrcParms**
Parameters of a connection established by this LE Client.

**DestParms**
Parameters of a connection received by this LE Client.

**NegoParms**
Parameters of a connection established by the LE Client for which the QoS Negotiation was used.

**RetryParms**
Parameters of a connection established by this LE Client after failing at least once.

#### Example:

```
LEC 1 QoS+ li vcc

                          LEC VCC Table
                          =============

                                                                        Burst
Conn   Conn              Conn          VCC    PCR    SCR   QoS   Size    Parameters
Index  Handle VPI   VCI   Type  Status  Type  (kbps) (kbps) Class (cells) Descriptor
-------------------------------------------------------------------------------------
   2)     69    0   535  Cntrl  Ready  BstEft 155000 155000    0      0   SrcParms
   3)     71    0   537  Cntrl  Ready  BstEft      0      0    0      0   DestParms
   4)     72    0   538  Mcast  Ready  BstEft 155000 155000    0      0   SrcParms
   5)     74    0   540  Mcast  Ready  BstEft      0      0    0      0   DestParms
   6)     78    0   544  Data   Ready  ResvBW  25000   7000    1     95   DestParms

LEC 1 QoS+
```

# Chapter 73. Using IP Security

Packets sent using the Internet Protocol (IP) can be made secure by using the IP Security feature of the 2216. This protection is provided by processes called authentication and encryption.

**Note:** Encryption support is optional and must be added to your software load using the **load add** command. See “Load” on page 88.

Security, as defined by RFC 1825-Security Architecture for the Internet Protocol, consists of these properties:

**Authentication**
Knowing that the data received is the same as the data that was sent and that the claimed sender is, in fact, the actual sender.

**Integrity**
Ensuring that data is transmitted from source to destination without undetected alteration.

**Confidentiality**
Communicating in such a way that the intended recipients know what was being sent but unintended parties cannot determine what was sent.

**Non-repudiation**
Communicating so that the receiver can prove that the sender did, in fact, send certain data even though the sender might later deny ever having sent that data.

The IP Security feature of the 2216 provides three of these properties: authentication, integrity, and confidentiality.

## Secure Tunnels

To protect the data sent to another host, router, or firewall, you can configure a secure tunnel. An IP secure (IPsec) tunnel is a two-way logical connection to the remote host, router, or firewall over which protected IP packets are transmitted. The IP Authentication Header (AH) and the IP Encapsulation Security Payload (ESP) are techniques that use special IP headers with authentication and encryption to ensure the security of the tunnel.

A secure tunnel is identified by many parameters, such as the tunnel ID and the address of the destination host at the far end of the tunnel. IP security is created on the 2216 by manually configuring a secure tunnel for each IP route that must be made secure. Each set of parameters specified creates one secure tunnel.

**Note:** For each secure tunnel, the parameters in the following list must match at each end of the secure tunnel; that is, the sender and the receiver must be configured with the same value:

- AH algorithm and AH authentication keys (See “Configuring the Algorithms” on page 870.)
- ESP encryption algorithm and ESP encryption and decryption keys (See “Configuring the Algorithms” on page 870.)

- Security parameters indexes (SPIs) (See “Security Associations”.)

## Tunnel Policy

A secure tunnel is configured with a tunnel policy that consists of one of these selections: AH, ESP, AH-ESP, or ESP-AH.

When both AH and ESP are configured, the following relationships apply:

- The policy AH-ESP means that for outbound packets, encryption is configured to run before authentication. In this case, inbound packets are checked by AH authentication first. Only the packets that are passed by AH authentication are forwarded to ESP for decryption.
- The policy ESP-AH means that for outbound packets, authentication is configured to run before encryption. In this case, inbound packets are decrypted by ESP first. Only the packets that are successfully decrypted are forwarded to AH authentication.

## Security Associations

Security associations (SAs) are one-way security connections that can use either AH or ESP to protect connection traffic. Two security associations or an SA bundle is configured for each secure tunnel—one outbound and one inbound. Each security association is identified by its own security parameters index (SPI), which is an arbitrary 32-bit value.

## Transport Mode and Tunnel Mode

Transport mode or tunnel mode is configured for each secure tunnel. Transport mode or tunnel mode determines the way in which AH or ESP handles the IP packets. Tunnel mode is the default. Transport mode is allowed only when the router is acting as a host. Tunnel mode is required if the router is acting as a security gateway.

### Modes Using AH

In transport mode, the AH is inserted after the IP header and before the header of an upper-layer protocol, such as TCP or UDP. In this mode, AH authenticates the upper-layer protocol header and the contents of the IP packet, except for the mutable fields in the IP header (such as time-to-live [TTL], checksum, fragment flag, fragment offset, and type of service [TOS]).

In tunnel mode, the AH is followed immediately by an entire IP packet and a new IP header is created and placed in front of the AH. The IP header of the packet being tunnelled (called the inner IP header) carries the ultimate source and destination addresses of the packet. The new IP header (called the outer IP header) can contain the addresses of security gateways, which are the tunnel endpoints. The AH protects the entire new packet, both the new IP header and the IP packet being tunnelled, except for the mutable fields in the new IP header.

### Modes Using ESP

In transport mode using ESP, the payload data field contains upper-layer protocol data, such as TCP or UDP data. The ESP encrypts the upper-layer protocol data

(and the ESP trailer, for IP security version 2). If authentication is used, the ESP header, the upper-layer protocol data, and the ESP trailer are authenticated.

In tunnel mode, the Payload Data field contains an entire IP packet and a new IP header is created and placed in front of the ESP. The IP header of the packet being tunnelled (called the inner IP header) carries the ultimate source and destination addresses of the packet while the new IP header (called the outer IP header) contains the addresses of security gateways. The ESP encrypts the tunnelled IP packet (and the ESP trailer, for IP security version 2). If ESP authentication is used, the ESP header, the tunnelled IP packet, and the ESP trailer are authenticated.

## IP Authentication Header (AH)

AH is described in draft-ietf-ispe-auth-header-05 Authentication Header. This header holds authentication data for the IP datagram. The sender of the datagram uses a cryptographic authentication function that relies upon a secret authentication key. This cryptographic authentication function is applied to the contents of the datagram.

### AH Authentication Algorithms

A secure tunnel that uses the AH tunnel policy must use one of these two authentication algorithms:

- HMAC-MD5 IP Authentication with Replay Prevention
- HMAC-SHA-1 IP Authentication with Replay Prevention

Both of these algorithms combine a keyed message authentication using cryptographic hash functions (abbreviated as HMAC) with replay prevention. Replay prevention, which is optional, uses a sequence number provided in the AH to verify that this packet has not been received before. Replay prevention is used to protect the receiver from denial-of-service attacks, where the same packets are repeatedly sent to the receiver. The router can become so busy processing the duplicate packets that it cannot process legitimate traffic. A sliding window is used to store enough sequence numbers to determine whether this sequence number has been received before.

## IP Encapsulating Security Payload (ESP)

ESP is described in draft-ietf-ipsec-esp-v2-04 Encapsulating Security Payload. ESP encrypts part or all of the IP packet to give you confidentiality as well as authentication and integrity. In ESP, the authentication function is optional.

### ESP Authentication Algorithms

The authentication algorithms available for ESP authentication are the same as for AH. See “AH Authentication Algorithms” for more information.

### ESP Encryption Algorithms

To configure ESP, you must choose one of three encryption algorithms:

- Data Encryption Standard in Cipher Block Chaining Mode (DES-CBC)
- Commercial Data Masking Facility (CDMF)
- Triple DES (3DES)

**Note:** The ESP encryption algorithms are subject to U.S. export laws. If your 2216 does not allow you to configure some or all of these algorithms, sale of those algorithms may be prohibited in your country. Check with your IBM representative for more information.

# Configuring the Algorithms

Depending upon the tunnel policy, algorithms are configured as shown in Table 132.

*Table 132. Algorithms Configured with Various Tunnel Policies*

| Tunnel Policy | Algorithms |
|---|---|
| AH, AH-ESP, or ESP-AH | • Local AH Authentication Algorithm—Required<br>• Remote AH Authentication Algorithm—Optional |
| ESP, AH-ESP, or ESP-AH | • Local Encryption Algorithm—Required<br>• Remote Encryption Algorithm—Optional<br>• Local ESP Authentication Algorithm—Optional<br>• Remote ESP Authentication Algorithm—Optional<br>**Note:** If your software load does not include encryption, you will not see encryption-related parameters. |

Local algorithms are applied to outbound packets and remote algorithms to inbound packets. The values for the remote algorithms are optional because each remote algorithm will take the value of the corresponding local algorithm as the default. The local ESP authentication algorithm is optional because authentication as part of ESP is an optional function.

The local algorithms configured by the sender for a particular secure tunnel must match the remote algorithms configured by the receiver at the far end of the secure tunnel. For example, if the sender tunnel policy is AH and the AH local authentication algorithm is HMAC-MD5, the receiver must have AH configured as one of its tunnel policies and the receiver's AH remote authentication algorithm must be HMAC-MD5.

## Configuring Keys

For each algorithm configured, a key must be configured as well. Each key must match the key for the same algorithm in the host at the far end of the tunnel. For example, if the local encryption key for outbound packets is 0098B1C588A109D5, the remote encryption key for inbound packets in the host at the far end of the secure tunnel must also be configured as the same number. See the descriptions of the keys in the **add tunnel** command in "Chapter 74. Configuring and Monitoring IP Security" on page 877 for more information.

# Example: Configuring an IPsec Tunnel

The network shown in Figure 64 on page 871 provides an example of an IPsec tunnel that connects a router with IPsec to a router with both IPsec and Network Address Translation (NAT).

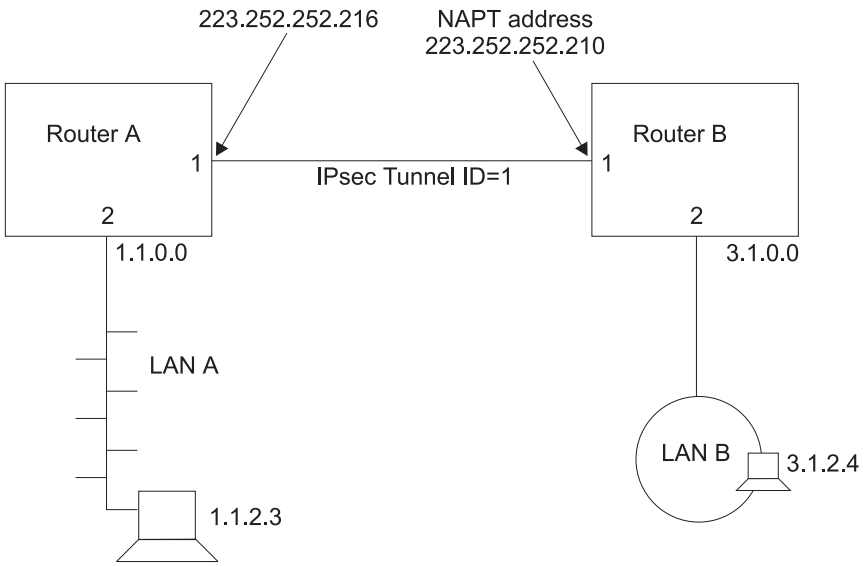



*Figure 64. Network with IPsec and NAT*

In this network, an IPsec tunnel with the IPsec tunnel ID 1 has been configured from IP address 223.252.252.216 in Router A to IP address 223.252.252.210 in Router B. Router A is configured for IPsec. Router B is configured for both IPsec and NAT. The following sections describe the process of configuring this network.

**Note:** If you do not plan to use NAT in your network, you will be more interested in Router A than Router B. However, reading over the description of configuring Router B can help you better understand the relationships between the parameters at each end of the IPsec tunnel.

## Configuring Router A (IPsec Only)

First, follow these steps to configure Router A.

- Create the IPsec tunnel.
- Create one outbound and one inbound packet filter on the router interface that is the endpoint of the IPsec tunnel.
- Create access control rules for the packet filters.
- Reset IPsec.
- Reset IP.

***Creating the IPsec Tunnel for Router A:*** The following example shows how to configure the IPsec tunnel 1 for Router A.

```
Config> feature ipsec
IP Security feature user configuration
IPsec config> add tunnel
IPsec Tunnel ID (1 - 65535) [1]
Tunnel Name (optional)? tunnelone
Tunnel Lifetime, in minutes (0-525600)[46080]?
Tunnel Encapsulation Mode (TUNN or TRANS) [TUNN]?
Tunnel Policy (AH, ESP, AH-ESP, ESP-AH) [AH-ESP]? AH
Local IP Address [1.1.1.1]? 223.252.252.216
Local Authentication SPI (256-65535)[256]?
Local Authentication Algorithm (HMAC-MD5, HMAC-SHA)[HMAC-MD5]?
Local Authentication Key (32 characters) in Hex (0-9,a-f,A-F):
Enter Local Authentication Key again (32 characters) in Hex (0-9,a-f,A-F):
Remote IP Address [0.0.0.0]? 223.252.252.210
Remote Authentication SPI (1-65535) [256]?
Remote Authentication Algorithm (HMAC-MD5, HMAC-SHA)[HMAC-MD5]?
```

```
Remote Authentication Key (32 characters) in Hex (0-9,a-f,A-F):
Enter Remote Authentication Key again (32 characters) in Hex (0-9,a-f,A-F):
Enable replay prevention? [No]:
Do you wish to enable this tunnel? [Yes]:
Ipsec config>
```

As you can see from this example, you are prompted for the parameters that you need to provide. The configuration of an ESP, AH-ESP, or ESP-AH secure tunnel calls for similar parameters.

**Note:** The values of the keys are not displayed when they are entered. Therefore, they are not visible in this example. If the keys for HMAC-MD5 authentication were visible, you would see 32 hex characters. For example, a key could have a value such as X'1234567890ABCDEF1234567890ABCDEF'.

***Configuring Packet Filters for Router A:*** After you have created the IPsec tunnel for Router A, you must set up two IP packet filters: one outbound packet filter and one inbound packet filter. The creation of the packet filter *out-router-A* is shown in the following example. Refer to the IP access control sections in the IP chapters in the *Protocol Configuration and Monitoring Reference, Vol. 1* for more information about configuring IP packet filters and access control rules.

```
*talk 6
Config> Protocol IP
Internet protocol user configuration
IP Config> set access-control on
IP Config> add packet-filter
Packet-filter name [ ]? out-router-A
Filter incoming or outgoing traffic? [IN]? OUT
Which interface is this filter for [0]? 1
IP Config>update packet-filter
Packet-filter name [ ]? out-router-A
Packet-filter 'out-router-A' Config>
```

In the same way, create an inbound packet filter for Router A on interface 1 in Router A called *in-router-A*. The packet filters are created on interface 1 because that is the endpoint of IPsec tunnel 1.

***Configuring Packet Filter Access Control Rules for Router A:*** The next step is to configure the packet filter access control rules. You should create two access control rules on the outbound packet filter *out-router-A* and two access control rules on the inbound packet filter *in-router-A*.

**Note:** Each IPsec tunnel must have an inbound and an outbound packet filter configured and two access control rules configured for each packet filter.

The access control rules on the outbound packet filter perform these functions:

- One access control rule defines the range of the source and destination addresses of the packets to be passed into the IPsec tunnel.
- The other access control rule allows IPsec traffic to pass through the packet filter.

The access control rules on the inbound packet filter perform these functions:

- One access control rule allows inbound IPsec traffic to pass through the packet filter.
- The other access control rule is an IPsec redundant check that examines the source and destination addresses of the packets that have been processed by IPsec. This access control rule assures that these source and destination addresses match the source and destination addresses of the packets that were outbound from the far end of the IPsec tunnel.

The first access control rule for *in-router-A* passes traffic over the IPsec tunnel by identifying the two endpoints of the IPsec tunnel. The protocol range 50 - 51 identifies IPsec.

```
IP Config> update packet-filter
Packet-filter name [ ]? in-router-A
Packet-filter 'in-router-A' Config> add access
Enter type [E]? I
Internet source [0.0.0.0]? 223.252.252.210
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 223.252.252.216
Destination mask [255.255.255.255]?
Enter starting protocol number ([0] for all protocols) [0]? 50
Enter ending protocol number [50]? 51
(Enable logging? (Yes or [No]):
Packet-filter 'in-router-A' Config>
```

The second access control rule for *in-router-A* checks the source and destination addresses of IPsec-processed packets on Router A to confirm that they are the same as the source and destination addresses of packets sent from Router B. This extra check on the security of the IPsec tunnel is redundant because the outbound packet filter on Router A should never pass packets with a source and destination address that does not match the source and destination address expected on the inbound packets at Router B. However, it is recommended in the IETF security architecture draft.

**Note:** Because Router B is using NAT, Router A does not have access to Router B's 3.1.0.0 addresses. For this reason, the second access control rule for *in-router-A* uses the address 223.252.252.210 rather than subnet 3.1.0.0 as the remote source address.

```
Packet-filter 'in-router-A' Config> add access
Enter type [E]? IS
Internet source [0.0.0.0]? 223.252.252.210
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 1.1.0.0
Destination mask [255.255.255.255]? 255.255.0.0
Enter starting protocol number ([0] for all protocols) [0]?
Enter IPsec Tunnel ID [1]?
(Enable logging? (Yes or [No]):
Packet-filter 'in-router-A' Config> exit
```

If you want all packets that do not match any access control rule to be passed rather than dropped, you can configure an inclusive wildcard access control rule to pass these packets. However, this access control rule invalidates the second inbound access control rule on the inbound packet filter because it passes the packets that the access control rule is designed to drop. The following example shows such an access control rule:

```
Packet-filter 'in-router-A' Config> add access
Enter type [E]? I
Internet source [0.0.0.0]?
Source mask [255.255.255.255]? 0.0.0.0
Internet destination [0.0.0.0]?
Destination mask [255.255.255.255]? 0.0.0.0
Enter starting protocol number ([0] for all protocols) [0]?
Enable Logging (Yes or [No]):
Packet-filter 'in-router-A' Config> exit
```

Next, configure the first access control rule for packet filter *out-router-A*. This access control rule passes packets from subnet 1.1.0.0 to the destination address 223.252.252.210 in Router B.

```
IP Config> update packet-filter
Packet-filter name [ ]? out-router-A
Packet-filter 'out-router-A' Config> add access
Enter type [E]? IS
Internet source [0.0.0.0]? 1.1.0.0
Source mask [255.255.255.255]? 255.255.0.0
Internet destination [0.0.0.0]? 223.252.252.210
```

```
Destination mask [255.255.255.255]?
Enter starting protocol number ([0] for all protocols) [0]?
Enter IPsec Tunnel ID [1]?
(Enable logging? (Yes or [No]):
Packet-filter 'out-router-A' Config>
```

The second access control rule for *out-router-A* allows packets to pass between the two ends of the IPsec tunnel.

```
Packet-filter 'out-router-A' Config> add access
Enter type [E]? I
Internet source [0.0.0.0]? 223.252.252.216
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 223.252.252.210
Destination mask [255.255.255.255]?
Enter starting protocol number ([0] for all protocols) [0]? 50
Enter ending protocol number [50]? 51
(Enable logging? (Yes or [No]):
Packet-filter 'out-router-A' Config>
```

As with the other packet filters, you may want to configure a wildcard access control rule for *out-router-A* to pass traffic that does not match any access control rules.

***Resetting IPsec and IP on Router A:*** After you complete your IPsec configuration, use the **reset ipsec** command in Talk 5 to reload SRAM with the new IPsec configuration that you created in Talk 6. The **reset ipsec** command does not affect any IP configuration. Then, use the **reset ip** command in Talk 5 to dynamically reset IP within the router. Alternatively, to reset each component, you can restart the router. It is necessary to reset IPsec and IP or to restart the router to assure that the packet filters and access rules are reloaded. Otherwise, your configuration may not be correctly supported on the interface. See “Chapter 74. Configuring and Monitoring IP Security” on page 877 and the **reset ip** command in the *Protocol Configuration and Monitoring Reference, Vol. 1* for more information.

## Configuring Router B (IPsec and NAT)

IPsec tunnel 1 has an endpoint on interface 1 in Router B. Router B will be configured for both IPsec and for NAT. When NAT is configured, you use the outbound packet filter on the router to pass outbound packets through NAT translation and IPsec encapsulation. The inbound packets pass IPsec for decryption first and then are passed to NAT for translation.

Follow these steps to configure Router B.

- Configure NAT.
- Create the IPsec tunnel.
- Create one outbound and one inbound packet filter on the router interface that is the endpoint of the IPsec tunnel.
- Create access control rules for the packet filters.
- Reset IPsec.
- Reset NAT.
- Reset IP.

The configuration of NAT in Router B is not discussed here. See “Chapter 75. Using Network Address Translation” on page 891 and “Chapter 76. Configuring and Monitoring Network Address Translation” on page 899 for information about configuring NAT. This example assumes that NAT has been configured and that the NAPT address 223.252.252.210 is also the endpoint of the IPsec tunnel. The NAT

private address pool in this example is 3.1.0.0 with the subnet 255.255.0.0. Inbound traffic arriving from IPsec tunnel 1 will be processed by IPsec, then passed to NAT for translation to one of these addresses.

**Notes:**

1. In this example, the IPsec tunnel endpoint address and the NAPT address are the same. However, in cases like this, when IPsec and NAT are used together, the address of the IPsec tunnel endpoint can be any valid IP address, not necessarily the NAPT address or one of the NAT public addresses.
2. If you are not concerned with NAT, you can regard the address 223.252.252.210 as the endpoint of IPsec tunnel 1 and the address range 3.1.0.0 simply as the address range of packets to be passed to IPsec.

***Creating the IPsec Tunnel for Router B:*** Within Router B, the same IPsec tunnel that was configured for Router A, IPsec tunnel 1, must be configured. The local IP address of this tunnel in Router B is 223.252.252.210 and the remote IP address is 223.252.252.216. All other IPsec tunnel parameters must match the parameters that were configured for Router A.

***Configuring Packet Filters for Router B:*** As you did for Router A, configure an inbound packet filter (*in-router-B*) and an outbound packet filter (*out-router-B*) on interface 1, which is the interface in Router B that is the endpoint of the IPsec tunnel 1.

***Configuring Packet-Filter Access Control Rules for Router B:*** First, configure the first inbound access control rule for the inbound packet filter *in-router-B* on Router B. This access control rule identifies the two endpoints of the IPsec tunnel and allows Router B to receive packets from the tunnel. This packet filter *in-router-B* is type inclusive (I).

```
IP Config> update packet-filter
Packet-filter name [ ] in-router-B
Packet-filter 'in-router-B' Config> add access
Enter type [E]? I
Internet source [0.0.0.0]? 223.252.252.216
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 223.252.252.210
Destination mask [255.255.255.255]?
Enter starting protocol number ([0] for all protocols) [0]? 50
Enter ending protocol number [50]? 51
Enable logging? (Yes or [No]):
Packet-filter 'in-router-B' Config>
```

Next, you can add the second access control rule to *in-router-B*.

This extra check on the security of the IPsec tunnel is redundant in IPsec However, this additional access control rule is required by NAT. Note that the access control rule is type I, N, and S.

```
Packet-filter 'in-router-B' Config> add access
Enter type [E]? INS
Internet source [0.0.0.0]? 1.1.0.0
Source mask [255.255.255.255]? 255.255.0.0
Internet destination [0.0.0.0]? 223.252.252.210
Destination mask [255.255.255.255]?
Enter starting protocol number ([0] for all protocols) [0]?
Enter IPsec Tunnel ID [1]?
Enable logging? (Yes or [No]):
Packet-filter 'in-router-B' Config>
```

If you want all packets that do not match any access control rule to be passed rather than dropped, you can configure an inclusive wildcard access control rule for *in-router-B* to pass these packets. However, this access control rule invalidates the

second inbound access control rule on the inbound packet filter because this access control rule passes the packets that the second access control rule is designed to drop.

Next, configure an access control rule on *out-router-B* to pass outbound packets from subnet 3.1.0.0 to NAT for translation and then to IPsec for processing and transmission through IPsec tunnel 1. This access control rule is type I, N, and S.

```
Packet-filter name [ ]? out-router-B
Packet-filter 'out-router-B' Config> add access
Enter type [E]? INS
Internet source [0.0.0.0]? 3.1.0.0
Source mask [255.255.255.255]? 255.255.0.0
Internet destination [0.0.0.0]? 1.1.0.0
Destination mask [255.255.255.255]? 255.255.0.0
Enter starting protocol number ([0] for all protocols) [0]?
Enter IPsec Tunnel ID [1]?
Enable logging? (Yes or [No]):
Packet-filter 'out-router-B' Config>
```

Now, for *out-router-B*, create an inclusive access control rule to let packets that have been processed by IPsec pass through IPsec tunnel 1.

```
Packet-filter 'out-router-B' Config> add access
Enter type [E]? I
Internet source [0.0.0.0]? 223.252.252.210
Source mask [255.255.255.255]?
Internet destination [0.0.0.0]? 223.252.252.216
Destination mask [255.255.255.255]?
Enter starting protocol number ([0] for all protocols) [0]? 50
Enter ending protocol number [50]? 51
(Enable logging? (Yes or [No]):
Packet-filter 'out-router-B' Config>
```

For *out-router-B*, create an inclusive wildcard access control rule if you wish to pass rather than drop packets that do not match either of the two access control rules, for example, traffic not destined for IPsec tunnel 1.

***Resetting NAT, IPsec, and IP on Router B:*** Before the NAT and IPsec functions will work and the IP access control rules are activated, NAT, IPsec, and IP have to be reset. Use the talk 5 **reset NAT** and **reset IPsec** commands to reset NAT and IPsec. See “Chapter 76. Configuring and Monitoring Network Address Translation” on page 899 for more information about resetting NAT and “Resetting IPsec and IP on Router A” on page 874 for information about resetting IPsec. After NAT and IPsec are reset, use the talk 5 **reset IP** command to reset IP. Alternatively, to reset each component, you can restart the router.

# Chapter 74. Configuring and Monitoring IP Security

This chapter describes how to configure and monitor IP security and how to use the IP security monitoring commands. It includes the following sections:

- "Accessing the IP Security Configuration Environment"
- "IP Security Configuration Commands"
- "Accessing the IP Security Monitoring Environment" on page 884
- "IP Security Monitoring Commands" on page 884

**Note:** If you create an IPsec tunnel to transport TN3270, APPN-ISR, or APPN-HPR traffic and you plan to prioritize that traffic using BRS, you need to use the IPv4 precedence bit setting feature of BRS. See "Using IP Version 4 Precedence Bit Processing for SNA Traffic in IP Secure Tunnels and Secondary Fragments" on page 687 for more information.

## Accessing the IP Security Configuration Environment

To access the IP Security configuration environment, enter the following command at the `Config>` prompt:

```
Config> feature ipsec
IP Security feature user configuration
IPsec config>
```

## IP Security Configuration Commands

This section describes the IP security configuration commands. Enter these commands at the `IPsec config>` prompt.

*Table 133. IP Security Configuration Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Add tunnel | Adds a secure tunnel. |
| Change tunnel | Changes a secure tunnel configuration parameter values. |
| Delete tunnel | Deletes a secure tunnel. |
| Disable | Disables all IP Security processing in a secure manner (packets that match the packet filters are dropped), disables all IP Security processing in a nonsecure manner (packets that match the packet filters are passed), or disables a secure tunnel. |
| Enable | Enables all IP Security processing, or enables a secure tunnel. |
| List | Lists information about global IP Security information, or information about defined tunnels. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Add Tunnel

Use the **add tunnel** command to add the parameters to define an IPsec tunnel.

**Note:** If these parameters are used, they are the same for AH, ESP, AH-ESP, and ESP-AH tunnel policies:

- Local SPI
- Local authentication algorithm
- Local authentication key
- Remote SPI
- Remote authentication algorithm
- Remote authentication key

**Syntax:**

**add tunnel...**

**tunnel-id**
Required number that specifies the identifier of the secure tunnel to be added. Each tunnel id must be unique within the 2216.

**Valid values**: 1 - 65536

**Default value**: none

**tunnel-name**
Optional parameter to label the tunnel. It must be unique within the 2216.

**Valid values**: up to 15 characters; first character must be a letter; no blanks can be used.

**Default value**: none

**lifetime**
Time in minutes that the tunnel can be active. The value 0 indicates that the tunnel lifetime never expires.

**Valid Values**: 0 - 525600 (0 = no expiration; 525600 = 365 days)

**Default Value**: 46080 (32 days)

**encapsulation-mode**
The manner in which the IP packet is encapsulated. In tunnel mode, the entire IP packet is encapsulated and a new IP header is created; in transport mode, the IP header is not encapsulated. If one end of the secure tunnel is a router, then tunnel mode ***must*** be used, according to the Internet Engineering Task Force (IETF) security architecture draft.

**Valid Values**: tunnel (*TUNN*) or translate (*TRANS*)

**Default Value**: tunnel (*TUNN*)

**tunnel-policy**
One of the four choices that define the tunnel policy: IP Authentication Header (AH), IP Encapsulating Security Payload (ESP), or combinations of these protocols (AH-ESP and ESP-AH). In AH-ESP, ESP encryption is run first on the outbound packets; in ESP-AH, AH authentication is run first on the outbound packets. Some parameters are unique either to ESP or AH. The encryption parameters are configured only if ESP, AH-ESP, or ESP-AH is selected; the authentication parameters are configured only if AH, AH-ESP, or ESP with authentication is selected.

**Valid Values**: AH, ESP, AH-ESP, ESP-AH

**Default Value**: AH-ESP

**local-IP-address**
IP address for this end of the tunnel.

**Valid Values**: a valid IP address that has been configured either for an interface or as the internal address of the 2216.

**Default Value**: 1.1.1.1

**local-spi**
A security association is a one-way security connection that uses AH or ESP to protect connection traffic. The security parameters index (SPI) is an arbitrary 32-bit value that uniquely identifies one of the two security associations (inbound or outbound) associated with this secure tunnel. This parameter, which is required, identifies the SPI expected in this tunnel for inbound packets received at the local end of the tunnel. This value cannot match the local SPI of another tunnel with the same local IP address. Regardless of the tunnel policy (ESP, AH, AH-ESP, or ESP-AH), only one local SPI is configured for inbound traffic for one IP secure tunnel.

**Valid Values**: 256 - 65535

**Default Value**: 256

**local-encryption-algorithm**
The encryption algorithm used for ESP on outbound packets sent from the local router, which is required when configuring ESP. This algorithm must match the encryption used by the workstation at the other end of the tunnel. In some countries, some or all of these algorithms may be unavailable because of U.S. export rules.

**Valid Values**: DES-CBC, CDMF, or 3DES

**Default Value**: DES-CBC

**local-encryption-key**
The key or keys used with the local ESP encryption algorithm. They must match the equivalent keys that are configured in the opposite end of the secure tunnel.

**Valid Values**:

- For DES-CBC: 16 hex characters (0 - 9, a - f, A - F)
- For CDMF: 16 hex characters (0 - 9, a - f, A - F)
- For 3DES: three separate keys, none of which is the same, each one 16 hex characters (0 - 9, a - f, A - F)

**Default Value**: none

**padding-for-local-encryption**
Size in bytes of additional padding that is added to outbound ESP packets. Additional padding may be used to disguise the size of the IP packets being encrypted when the encryption algorithm results in an encrypted packet that is the same size as the original packet. ESP padding values must be a multiple of 8. If a value that is not divisible by 8 is configured, that value is rounded up to the next value that is divisible by 8.

**Valid Values**: 0 - 120

**Default Value**: 0

**local-ESP-authentication**
Selects local ESP authentication, if desired.

**Valid Values**: Yes or No

**Default Value**: Yes

**local-authentication-algorithm**
The authentication algorithm used on outbound packets. This is an optional parameter for ESP and will not be required unless you select ESP authentication. For AH , AH-ESP, or ESP-AH, this parameter is required. The authentication algorithm used must match the remote authentication algorithm used at the far end of the IPsec tunnel.

**Valid Values**: HMAC-MD5 or HMAC-SHA

**Default Value**: HMAC-MD5

**local-authentication-key**
The key used with the local authentication algorithm. It must match the equivalent key that is configured in the opposite end of the IPsec tunnel. It is required if the policy is AH, AH-ESP, or ESP-AH, or if the policy is ESP and the local ESP authentication algorithm has been configured.

**Valid Values**:

- for HMAC-MD5: 32 hex characters (0 - 9, a - f, A - F)
- for HMAC-SHA: 40 hex characters (0 - 9, a -f, A - F)

**Default Value**: none

**remote-IP-address**
IP address for the remote end of the tunnel. This is a required parameter.

**Valid Values**: a valid IP address

**Default Value**: 1.1.1.3

**remote-spi**
A security association is a one-way security connection that uses AH or ESP to protect connection traffic. The security parameters index (SPI) is an arbitrary 32-bit value that uniquely identifies one of the two security associations (inbound or outbound) associated with this secure tunnel. This parameter, which is required, identifies the SPI expected in ESP or AH for outbound packets destined for the remote host. This value cannot match the remote SPI of another tunnel with the same remote IP address. Regardless of the tunnel policy (ESP, AH, AH-ESP, or ESP-AH), only one local SPI is configured for outbound traffic for one IPsec tunnel.

**Valid Values**: 1 - 65535

**Default Value**: 256

**remote-encryption-algorithm**
The decryption algorithm used on inbound packets received from the remote host.

**Valid Values**:DES-CBC, CDMF, or 3DES

**Default Value**: value of the local encryption algorithm

**remote-encryption-key**
The key or keys used with the remote ESP encryption algorithm. They must match the equivalent keys that are configured in the opposite end of the secure tunnel.

**Valid Values**:

- For DES-CBC: 16 hex characters (0 - 9, a - f, A - F)
- For CDMF: 16 hex characters (0 - 9, a - f, A - F)

- For 3DES: three separate keys, none of which matches, each 16 characters in hex (0 - 9, a - f, A - F)

**Default Value**: none

**verification-of-remote-encryption-padding**
Determines whether the size of the encryption padding on received packets should be verified.

**Valid Values**: Yes or No

**Default Value**: No

**padding-for-remote-encryption**
Size in bytes of additional padding that is expected in received ESP packets. This parameter is required and valid only if the value of *verification-of-remote-encryption-padding* is Yes. ESP padding values must be a multiple of 8. If a value that is not divisible by 8 is configured, that value will be rounded up to the next value that is divisible by 8.

**Valid Values**: 0 - 120

**Default Value**: 0

**remote-ESP-authentication**
Selects remote ESP authentication for inbound packets, if desired.

**Valid Values**: Yes or No

**Default Value**: Yes

**remote-authentication-algorithm**
The authentication algorithm used for inbound packets. This is an optional parameter for ESP and will not be required unless you select ESP authentication. For AH or combinations of AH and ESP (AH-ESP or ESP-AH), this parameter is required. The authentication algorithm used must match the local authentication algorithm used at the far end of the IPsec tunnel.

**Valid Values**: HMAC-MD5 or HMAC-SHA

**Default Value**: HMAC-MD5

**remote-authentication-key**
The key used with the remote authentication algorithm. It must match the equivalent key that is configured in the opposite end of the secure tunnel. It is required in AH, AH-ESP and ESP-AH and in ESP if the remote ESP authentication algorithm has been configured.

**Valid Values**:

- for HMAC-MD5: 32 hex characters (0 - 9, a - f, A - F)
- for HMAC-SHA: 40 hex characters (0 - 9, a -f, A - F)

**Default Value**: none

**enable-replay-prevention**
Specifies whether replay prevention is enabled. If replay prevention is enabled, the sequence numbers in the IP security headers are monitored to prevent duplicate packets from being processed by the tunnel receiver. The use of replay prevention is not recommended because the tunnel security association must be deactivated when a sender's sequence number counter reaches its limit. When this happens, manual intervention is required to restart the existing security association or create a new one.

In addition, if replay prevention is enabled and you reset IPsec using the **reset ipsec** command, you must make sure that IPsec is also reset on the router at the other end of the IPsec tunnel. This is necessary to re-initialize the sequence number at both ends of the tunnel. If IPsec is reset on one end of the tunnel and not on the other, it is possible that routers at each end of the tunnel will drop packets due to sequence number mismatch.

**Valid Values**: Yes or No

**Default Value**: No

**enable-tunnel**
Specifies whether this tunnel is enabled. The enabled tunnel will not filter packets until a packet filter has been configured to define the interface over which this IPsec tunnel will operate and IP has been reset or restarted on the 2216. You can use the **reset ip** command to reset IP.

**Valid Values**: Yes or No

**Default Value**: Yes

## Change Tunnel

Use the **change tunnel** command to change an IPsec tunnel parameter previously configured by the **add tunnel** command.

**Syntax:**

**change tunnel...** See the **add tunnel** command for a list of the parameters that can be changed.

## Delete Tunnel

Use the **delete tunnel** command to delete an IPsec tunnel.

**Syntax:**

**delete tunnel** *tunnel-id tunnel-name* **all**

**tunnel-id**
Specifies the identifier of the IPsec tunnel to be deleted.

**Valid Values**: 1 - 65536

**Default Value**: 1

**tunnel-name**
Specifies the name of the IPsec tunnel to be deleted.

**Valid Values**: any configured tunnel name

**Default Value**: none

**all** Specifies that all IPsec tunnels on this interface are to be deleted.

## Disable

Use the **disable** command to disable the IPsec tunnel or to disable all IPsec tunnels either in a secure manner (packets that match the IPsec filters are dropped) or an insecure manner (packets that match the IPsec filters are passed).

**Syntax:**

**disable** ipsec drop

ipsec pass

tunnel ...

**ipsec drop**
Disables IP security on the router in a secure manner. All IPsec tunnels will be disabled, but the secure tunnel information in packet filter rules is used to identify packets that match IPsec tunnel packet filters. The matching packets are dropped.

**ipsec pass**
Disables IP security on the router in a non-secure manner. All IPsec tunnels will be disabled. Packets that match IPsec tunnel packet filters are forwarded as ordinary traffic.

**tunnel** *tunnel-id* **all**
Disables IP security on a specified tunnel or on all tunnels.

**tunnel-id**
Specifies the identifier of the secure tunnel to be disabled.

**Valid Values**: 1 - 65536

**Default Value**: 1

**all** All tunnels.

## Enable

Use the **enable** command to enable the IP Security protocol on all interfaces or a single tunnel. You must enable ipsec globally on the router before the individually enabled IPsec tunnels become active.

**Syntax:**

**enable** ipsec

tunnel ...

**ipsec** Enables IP security throughout the router.

**tunnel** *tunnel-id* **all**
Enables IP security on a specified tunnel or on all tunnels.

**tunnel-id**
Specifies the identifier of the secure tunnel to be enabled.

**Valid Values**: 1 - 65536

**Default Value**: 1

**all** All tunnels.

## List

Use the **list tunnel** command to display the current IP Security configuration. Global tunnels include all tunnels in the router, both active and defined. All tunnels

include all tunnels configured on this interface, both active and defined. Active tunnels are those that are currently active; defined tunnels are defined but not active.

**Syntax:**

**list ...** all
global
tunnel
active *tunnel-id tunnel-name* all
defined *tunnel-id tunnel-name* all

**Example:**

```
IPsec config>list all

IPsec is ENABLED

Defined Manual Tunnels:

   ID         Name         Local IP Addr   Remote IP Addr   Mode    State
 ------  ---------------  ---------------  ---------------  -----  --------
      1  test                    1.1.1.1          2.1.1.1  TUNN   Enabled
      2  test2                   1.1.1.1          1.1.1.3  TRANS  Enabled

Tunnel Cache:

 ID     Local IP Addr   Remote IP Addr   Mode   Policy  Tunnel Expiration
-----  ---------------  ---------------  -----  ------  -----------------
    2          1.1.1.1          1.1.1.3  TRANS  ESP     ****************
    1          1.1.1.1          2.1.1.1  TUNN   AH      ****************
```

# Accessing the IP Security Monitoring Environment

To access the IP Security monitoring environment type **t 5** at the OPCON prompt (*):

```
* t 5
```

Then, enter the following command at the **+** prompt:

```
+ feature ipsec
IPsec>
```

# IP Security Monitoring Commands

This section describes the IP Security monitoring commands. Enter these commands at the IPsec> prompt.

*Table 134. IP Security Monitoring Commands Summary*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| Add tunnel | Dynamically adds a secure tunnel. |
| Change tunnel | Dynamically changes a secure tunnel configuration parameter values. |
| Delete tunnel | Dynamically deletes a secure tunnel. |

*Table 134. IP Security Monitoring Commands Summary (continued)*

| Command | Function |
|---|---|
| Disable | Dynamically disables all IP Security processing in a secure manner (matching packets are dropped), disables all IP Security processing in a nonsecure manner (matching packets are forwarded), or disables a particular secure tunnel. |
| Enable | Dynamically enables all IP Security processing, or enables a secure tunnel. |
| List | Lists information about global IP Security information, or information about active and defined tunnels. |
| Reset | Resets IP Security or resets a secure tunnel. This command reloads the configuration that was created in Talk 6. Resetting will override the values of parameters configured using Talk 5 with those that were configured using Talk 6. |
| Restart | Restarts IP Security or restarts a secure tunnel. This command reloads the configuration information that has been dynamically configured using Talk 5 commands. |
| Stats | Displays statistics for all tunnels or for an active tunnel. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## Add Tunnel

Dynamically adds a secure tunnel.

**Syntax:**

**add tunnel ...**

See the **add tunnel** command under “IP Security Configuration Commands” on page 877 for a description of the parameters.

## Change Tunnel

Dynamically changes a secure tunnel.

**Syntax:**

**change tunnel ...**

See the description of the **add tunnel** command under “IP Security Configuration Commands” on page 877 for a description of the parameters.

## Delete Tunnel

Use the **delete** command to dynamically delete a secure tunnel or all secure tunnels.

**Syntax:**

**delete tunnel** *tunnel-id tunnel-name* **all**

**tunnel-id**

Specifies the identifier of the IPsec tunnel to be deleted.

**Valid Values**: 1 - 65536

**Default Value**: 1

**tunnel-name**
Specifies the name of the IPsec tunnel to be deleted.

**Valid Values**: any configured tunnel name

**Default Value**: none

**all** Specifies that all IPsec tunnels on this interface are to be deleted.

## Disable

Use the **disable** command to dynamically disable the IP Security protocol on all interfaces or a single tunnel.

**Syntax:**

**disable** ipsec drop
ipsec pass
tunnel ...

**ipsec drop**
Disables IP security on the router in a secure manner. All IPsec tunnels will be disabled, but the secure tunnel information in packet filter rules is used to identify packets that match IPsec tunnel packet filters. The matching packets are dropped.

**ipsec pass**
Disables IP security on the router in a non-secure manner. All IPsec tunnels will be disabled. Packets that match IPsec tunnel packet filters are forwarded as ordinary traffic.

**tunnel** *tunnel-id* **all**
Disables IP security on a specified tunnel or on all tunnels.

**tunnel-id**
Specifies the identifier of the secure tunnel to be disabled.

**Valid Values**: 1 - 65536

**Default Value**: 1

**all** All tunnels.

## Enable

Use the **enable** command to dynamically enable the IP Security protocol on all interfaces or a single tunnel. You must enable ipsec globally on the router before the individually enabled IPsec tunnels become active.

**Note:** IPsec cannot be dynamically enabled if the router was restarted with IPsec disabled.

**Syntax:**

**enable** ipsec
tunnel ...

**ipsec** Enables IP security throughout the router.

**tunnel** *tunnel-id* **all**

**tunnel-id**
Specifies the identifier of the secure tunnel to be enabled.

**Valid Values**: 1 - 65536

**Default Value**: 1

**all** All tunnels.

## List

Use the **list** command to display the current IP Security configuration. Global tunnels include all tunnels in the router, both active and defined. All tunnels include all tunnels configured on this interface, both active and defined. Active tunnels are those that are currently active; defined tunnels are defined but not active.

**Syntax:**

**list ...** all
global
tunnel
active *tunnel-id tunnel-name* all
defined *tunnel-id tunnel-name* all

**Example:**

```
IPsec>li tunnel ?
ACTIVE
DEFINED
IPsec>li tunnel active
Enter the Tunnel ID, Tunnel Name, or 'ALL' [ALL]? all

Tunnel Cache:

 ID     Local IP Addr   Remote IP Addr   Mode   Policy  Tunnel Expiration
-----  ---------------  ---------------  -----  ------  ------------------
    2          1.1.1.1          1.1.1.3  TRANS  ESP     *****************
    1          1.1.1.1          2.1.1.1  TUNN   AH      *****************
```

## Reset

Use the **reset** command to dynamically reset IP security on the router or on a single tunnel. After you reset IPsec or the tunnels, be sure to use the **reset IP** command to reset the IP configuration. This is necessary to reload the access control information, such as packet filters and their access control rules. If you do not reset IP, the packet filters and access control rules may not support your new IPsec configuration.

Rebooting the router is an alternative to using the **reset** commands. However, rebooting the router takes it off the network for a time, whereas the **reset** commands interrupt only IP functions.

**Syntax:**

**reset** ipsec
tunnel *tunnel-id tunnel-name* all

**ipsec** Resets IP security on the 2216. IP security is temporarily disabled and then restarted. While IP security is disabled, any packets that are normally handled by IPsec tunnels are dropped until the reset is complete. Resetting IP security does not affect other functions on the 2216. This command activates the IP security configuration that was created using Talk 6. The Talk 6 IP security configuration overwrites the Talk 5 configuration.

**tunnel** Resets IP security on a specified tunnel. If the tunnel is disabled at the time of reset, the tunnel configuration is rebuilt from the SRAM configuration, but the tunnel remains disabled after the reset.

**tunnel-id**
Specifies the identifier of the secure tunnel to be reset.

**Valid Values**: 1 - 65536

**Default Value**: 1

**tunnel-name**
Specifies the name of the secure tunnel to be reset.

**Valid Values**: any configured tunnel name

**Default Value**: none

**all** All tunnels.

## Restart

Use the **restart** command to dynamically restart IP security on the router or on a single tunnel. This restarts the temporary configuration that was created using Talk 5. The Talk 6 IP security configuration does not overwrite the Talk 5 configuration.

**Syntax:**

**restart** ipsec
tunnel *tunnel-id tunnel-name* all

**ipsec** Restarts IP security on the 2216.

**tunnel** Restarts IP security on a specified tunnel.

**tunnel-id**
Specifies the identifier of the secure tunnel to be reset.

**Valid Values**: 1 - 65536

**Default Value**: 1

**tunnel-name**
Specifies the name of the secure tunnel to be reset.

**Valid Values**: any configured tunnel name

**Default Value**: none

**all** All tunnels.

## Stats

Use the **stats** command to display statistics about a specific tunnel or all tunnels. For example, the **stats** command shows packets sent and received.

**Syntax:**

**stats** *tunnel-id tunnel-name* **all**

**tunnel-id**

Specifies the identifier of the secure tunnel.

**Valid Values**: 1 - 65536

**Default Value**: 1

**tunnel-name**

Specifies the name of a secure tunnel that has been configured.

**Valid Values**: any configured tunnel name

**Default Value**: none

**all** Displays statistics about all tunnels configured on the 2216.

**Example:**

```
IPsec>stats
Enter the Tunnel ID, Tunnel Name, or 'ALL' [ALL]? all

                          Global IPSec Statistics
Received:
  total pkts   AH packets   ESP packets   total bytes    AH bytes     ESP bytes
  ----------   ----------   -----------   -----------   ----------   ----------
           0            0             0             0            0            0

Sent:
  total pkts   AH packets   ESP packets   total bytes    AH bytes     ESP bytes
  ----------   ----------   -----------   -----------   ----------   ----------
           0            0             0             0            0            0

Receive Packet Errors:
  total errs    AH errors   AH bad seq   ESP errors   ESP bad seq
  ----------   ----------   ----------   ----------   -----------
           0            0            0            0             0

Send Packet Errors:
  total errs    AH errors   ESP errors
  ----------   ----------   ----------
           0            0            0
```

# Chapter 75. Using Network Address Translation

Network Address Translation (NAT) and its extension Network Address and Port Translation (NAPT) can expand the number of IP addresses available to an organization and can prevent users in the public network from becoming aware of some of the addresses in the private network. NAT works by using public IP addresses to represent private IP addresses.

Public IP addresses are the valid addresses of hosts in the IP public network and they must be unique within the public network. If the public network is the Internet, the public IP addresses must be unique Internet addresses provided by the Network Information Center (NIC).

The private addresses are known to the router, but not to the public network. The addresses within each private network must be unique; however, the same address can be duplicated in two different private networks. The private addresses are assigned to hosts within stub networks. Stub networks are networks that have access to the public network through one router only.

NAT expands the number of available IP addresses in several ways:

- It allows each public address to represent multiple private addresses by rotating the use of the public addresses.
- It allows the duplication of addresses as long as each duplicate address is used in a different private network.
- It allows the network administrator to use any IP addresses in the private networks, instead of the NIC addresses that are becoming limited resources.

Using private addresses also hides these addresses from the outside world. This feature of NAT makes it useful as a type of firewall to protect the private addresses from being known.

**Important:** As stated in section 5.4 of the Internet Draft which defines NAT, “any application that carries (and uses) the IP address (and TCP/UDP port, in the case of NAPT) inside the application will not work through NAT...”. It should be noted that DLSw and XTP make decisions based on the end-point IP addresses — specifically which partner has the higher address. Since the application (such as DLSw or XTP) that is running through NAT thinks that its address is the private address, but the partner application in the other router thinks that the application’s address is the public address, incorrect decisions can be made.

See Figure 65 on page 892 for a drawing of a workstation in a stub network. In this example, the stub network consists of an IP subnet that has the IP address 10.33.96.0 with the subnet mask 255.255.255.0.

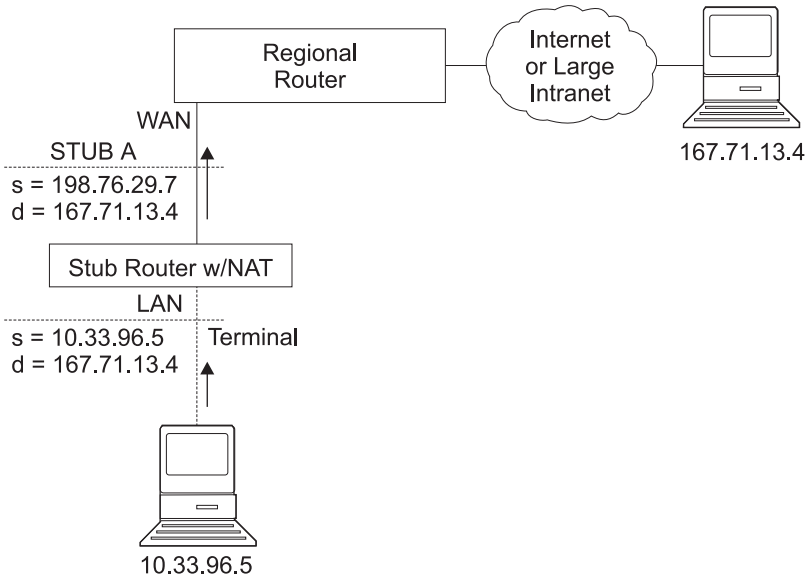



*Figure 65. Network Running NAT*

To use NAT, the network administrator assigns one or more public IP addresses to a public address pool in the 2216 and assigns a private IP address to each workstation in the stub network. The public IP addresses are assigned to a *reserve pool* and the private IP addresses are assigned to the *translate range*.

The NAT function first binds the private address of a station in the private network to one of the public addresses. Binding means that every packet with that private address will be translated to that public IP address when the packet is outbound. Inbound packets have the public IP address as their destination. NAT recognizes the public address, translates it to the private IP address, and forwards the packet. After traffic stops, the binding is maintained until a timer that you can set times out. At this time, NAT ends the binding and makes the public address available for reuse.

In this example, a packet is transmitted from sending private source address 10.33.96.5 to a destination address in the Internet, 167.71.13.4. NAT in the 2216 translates private address 10.33.96.5 to public address 198.76.29.7. This translation hides the private address 10.33.96.5 from the public network, so that no incoming packet is addressed directly to private address 10.33.96.5. Instead, incoming packets from 167.71.13.4 are addressed to public address 198.76.29.7. When the NAT router receives packets addressed to 198.76.29.7, NAT translates the destination public address to the private address 10.33.96.5 and forwards the packets.

## Network Address Port Translation

NAPT can be used only for TCP and UDP traffic. In NAPT, multiple private addresses can use a single public address simultaneously. While NAT maps one public address to one private address, NAPT maps the NAPT public address ***and*** the public port number to a private address and private port number. Only one NAPT address can be configured for each public address pool.

NAPT is configured simply by configuring one public address that will be used for NAPT traffic. The advantage of NAPT is that it can enable one address from the pool of public IP addresses to support many private IP addresses simultaneously.

## Static Address Mappings

Sometimes you may want to configure a station or server in the private network that can be directly accessed from the public network. In this case, you should make a static mapping of the private address of the station to a particular public address. All messages outbound from the private address are translated to the designated public address and all messages inbound for the designated public address are automatically forwarded to the associated private address. There are two kinds of static address mappings: NAT and NAPT.

### NAT Static Address Mapping

In a NAT mapping, all IP protocols can access the host. This is an example of the configuration of a NAT mapping:

| | |
|---|---|
| Private address | 10.1.1.2 |
| Private port | 0 |
| Public NAT address | 9.67.1.1 |
| Public port | 0 |

### NAPT Static Address Mapping

To specify a TCP or UDP application, you have the option to specify a NAPT mapping that includes a private well-known port. For NAPT static address mapping, a NAPT public address must be configured. For example, to configure a Telnet host at private address 10.1.1.1 to use the NAPT public address 9.67.1.2, the static mapping would be configured as follows:

| | |
|---|---|
| Private address | 10.1.1.1 |
| Private port | 23 |
| Public NAPT address | 9.67.1.2 |
| Public port | 23 |

The private and public ports are mapped to port 23, which is the well-known port for Telnet. Now, if the administrator also has an FTP server (well-known address 21) at the same private address 10.1.1.1 to map to the NAPT public address 9.67.1.2, that mapping can look like this:

| | |
|---|---|
| Private address | 10.1.1.1 |
| Private port | 21 |
| Public NAPT address | 9.67.1.2 |
| Public port | 21 |

The server at address 10.1.1.1 has the same NAPT public address (9.67.1.2) for both applications, but NAPT can distinguish between the two by using the different port numbers (23 and 21). However, NAPT cannot distinguish between two servers that use the same NAPT public address and have the same application and port

number. For example, if the NAPT public address and well-known port are the same for 10.1.1.3 port 21 as for 10.1.1.1 port 21, NAPT cannot tell whether to send incoming FTP traffic to server 10.1.1.3 or 10.1.1.1. To configure more than one server with the same NAPT address and application, you must use a port other than the well-known port at the server (for example, start the FTP daemon on port 200).

## Setting Packet Filters and Access Control Rules for NAT

In addition to identifying the range of private addresses to be translated by NAT or NAPT, the administrator must set up packet filters and access control rules for IP in the 2216. NAT configuration requires you to configure one inbound and one outbound packet filter on the interface that is connected to the public network. You need to configure one or more access control rules on the inbound packet filter and one or more access control rules on the outbound packet filter. The inbound filter access control rules pass inbound packets with the appropriate defined public addresses to NAT. The outbound filter access control rules pass outbound packets with the appropriate defined private addresses to NAT.

The access control rules that are applied for NAT have the access control rule types ***I*** and ***N*** for inclusive and NAT. Refer to the *Protocol Configuration and Monitoring Reference, Vol. 1* for information about configuring IP access controls.

**Note:** NAT can also be configured in conjunction with an IPsec tunnel. A sample of this configuration is found in “Configuring Packet Filter Access Control Rules for Router A” on page 872.

## Example: Configuration of NAT With IP Filters and Access Control Rules

This example shows how to configure NAT for the stub router in the network pictured in Figure 66 on page 895. See “Chapter 76. Configuring and Monitoring Network Address Translation” on page 899 for descriptions of the commands.

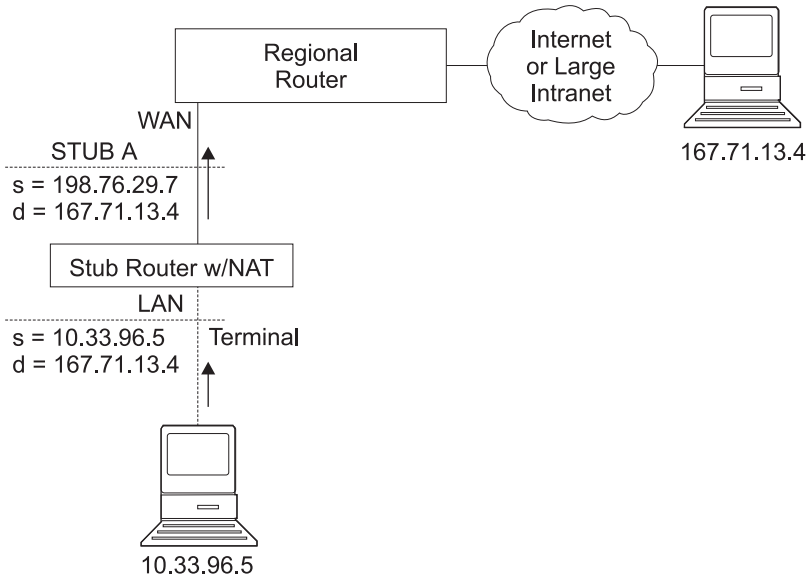



*Figure 66. Network Running NAT*

Follow this procedure:

1. Set up pools of public addresses for use by NAT and NAPT. To do this, use the **reserve** command.

   ```
   NAT config> reserve 198.76.29.7 255.255.255.0 6 pool1 198.76.29.7
   NAT config> reserve 198.76.29.15 255.255.255.0 3 pool1 0.0.0.0
   ```

   In this example, a pool called *pool1* is established. The NAPT address in the pool is 198.76.29.7. The addresses 198.76.29.13 and 198.76.29.14 are not available, so the pool is set up to exclude them. The parameters entered are: *public-address, mask, number-in-group, name, and napt-address*. The value 0.0.0.0 for the NAPT address means that none of the addresses in this group is the NAPT address. Use 0.0.0.0 for the NAPT address in all groups if you do not configure NAPT for the pool.
2. Use the **translate** command to establish the ranges of private addresses to be translated by the public addresses in pool1. The parameters entered are: *private-address, mask, and name*.

   ```
   NAT config> translate 10.33.96.0 255.255.255.0 pool1
   ```

3. Set up static mappings for stations inside the private network that are to be permanently mapped to one of the public addresses. The following commands identify one machine (10.33.96.5) that will receive any type of traffic from the public network. A second machine (10.33.96.4) is both a Telnet and an HTTP server. The parameters are *private-address, private-port-number, public-address, and public-port-number*. Note that the NAPT address for pool1 is used as the public address for the host that is configured with two port numbers.

   ```
   NAT config> map 10.33.96.5 0 198.76.29.8 0
   NAT config> map 10.33.96.4 23 198.76.29.7 23
   NAT config> map 10.33.96.4 80 198.76.29.7 80
   ```

4. Enable NAT.

   ```
   NAT config> enable NAT
   ```

5. Create two IP packet filters so that IP will pass packets to NAT. These are inbound and outbound packet filters for interface 0, which is the interface connected to the public network.

```
IP Config> add packet-filter outbound out-0 0
IP Config> add packet-filter inbound in-0 0
```

6. Use the **update** command to bring up the `packet-filter '`*`filter-name`*`' Config>` prompt. Add an access control rule for NAT to the inbound filter. Packets received over the public interface (net 0) that are destined for an address in NAT's reserved public address pool should be passed to NAT. NAT will replace the public address (and the public port if the packet is destined for the NAPT address) with the correct private address (and the private port if the packet is destined for the NAPT address). The 0.0.0.0 address and mask for the Internet source indicate that any source addresses from the public network will be passed to NAT.

```
IP Config>update packet-filter
Packet-filter name [ ]? in-0
Packet-filter 'in-0' Config> add access
Enter type [E]? IN
Internet source [0.0.0.0]?
Source mask [255.255.255.255]? 0.0.0.0
Internet destination [0.0.0.0]? 198.76.29.0
Destination mask [255.255.255.255]?255.255.255.0
Enter starting protocol number ([0] for all protocols) [0]?
Enable logging? (Yes or [No]):
Packet-filter 'in-0' Config>
```

   The range of addresses in the access control rule is greater than the range of addresses defined in pool1. If the address of the packet passed to NAT is in the range defined in the access control rule but is not one of the ones in the public address pool, NAT passes the packet back to IP unchanged.

7. If you wish the router to pass the packets that do not match the access control rule, rather than drop them, you can create a wildcard access control rule. The following example shows such an access control rule:

```
Packet-filter 'in-0' Config> add access
Enter type [E]? I
Internet source [0.0.0.0]? 0.0.0.0
Source mask [255.255.255.255]? 0.0.0.0
Internet destination [0.0.0.0]? 0.0.0.0
Destination mask [255.255.255.255]?0.0.0.0
Enter starting protocol number ([0] for all protocols) [0]?
Enable logging? (Yes or [No]):
Packet-filter 'in-0' Config>
```

8. Add an access control rule for NAT to the outbound packet filter. Packets to be forwarded from the net 0 interface that have a source address on the private network are identified so that IP can pass them to NAT. NAT replaces the private address with one of the public addresses in pool1.

```
Packet-filter 'out-0' Config> add access
Enter type [E]? IN
Internet source [0.0.0.0]? 10.33.96.0
Source mask [255.255.255.255]? 255.255.255.0
Internet destination [0.0.0.0]?
Destination mask [255.255.255.255]?0.0.0.0
Enter starting protocol number ([0] for all protocols) [0]?
Enable logging? (Yes or [No]):
Packet-filter 'out-0' Config>
```

   With this packet filter as with filter *in-0*, you can add a wildcard inclusive access control rule as the last access control rule if you plan to forward packets that do not match the access control rule.

9. You can use the **list packet-filter** *filter-name* command from the `IP Config>` prompt to check the accuracy and sequence of the access control rules in each packet filter.

10. Enable the access controls for IP.

```
IP Config> set access-control on
```

11. Reset IP and NAT using talk 5. Until now, you have created changes in the router configuration, but these changes have not affected the router. The reset commands for IP and NAT cause the router to read in the new configuration and run with the rules defined in the configuration.

```
NAT> reset NAT
IP> reset IP
```

# Chapter 76. Configuring and Monitoring Network Address Translation

This chapter describes the Network Address Translation (NAT) configuring and monitoring commands and includes the following sections:



## Accessing the Network Address Translation Configuration Environment

To access the NAT configuration environment, enter the following command at the `Config>` prompt:

```
Config> feature nat
Network Address Protocol user configuration
NAT config>
```

## Network Address Translation Configuration Commands

This section explains the Network Address Translation (NAT) configuration commands. To configure NAT, enter these commands at the `NAT config>` prompt.

*Table 135. NAT Configuration Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See "Getting Help" on page 10. |
| Change | Changes public IP address reserve pools, private address translate ranges, and static mappings. |
| Delete | Deletes public IP address reserve pools, private address translate ranges, and static mappings. |
| Disable | Disables NAT. |
| Enable | Enables NAT. |
| List | Lists information about the NAT configuration. |
| Map | Creates a static NAT or NAPT binding for a station or server. |
| Reserve | Creates a public IP address pool and appends addresses to that pool. |
| Reset | Causes the router to read in the NAT configuration and run according to the NAT rules that have been configured. |
| Set | Sets timeouts. |
| Translate | Identifies the private IP addresses to be translated by the NAT public address pool. |
| Exit | Returns you to the previous command level. See "Exiting a Lower Level Environment" on page 11. |

## Change

Use the **change** command to change public IP address reserve pools, private IP address translate ranges, and static mappings.

**Syntax:**

**change** reserve
translate
mappings

**reserve** *pools*
Provides prompts that enable you to change characteristics of any of the public IP address reserve pools (such as IP addresses and masks) .

**Valid Values**: An index number to identify the configured pool. This number is displayed when you enter the **list reserve pools** command.

**Default Value**: none

**translate** *ranges*
Provides prompts that enable you to change characteristics of any of the private IP address translate ranges (such as IP addresses and masks).

**Valid Values**: An index number to identify the configured translate range. This number is displayed when you enter the **list translate** command.

**Default Value**: none

**mappings**
Provides prompts that enable you to change characteristics of any of the static address mappings (such as IP addresses and ports).

**Valid Values**: An index number to identify the configured mapping. This number is displayed when you enter the **list mappings** command.

**Default Value**: none

## Delete

Use the **delete** command to delete public IP address reserve pools, private IP address translate ranges, and mappings.

**Syntax:**

**delete** reserve
translate
mappings

**reserve** *pools*
Provides prompts that enable you to delete any of the public IP address reserve pools.

**Valid Values**: An index number to identify the configured pool. This number is displayed when you enter the **list reserve pools** command.

**Default Value**: none

**translate** *ranges*
Provides prompts that enable you to delete any of the private IP address translate ranges.

**Valid Values**: An index number to identify the configured translate range. This number is displayed when you enter the **list translate** command.

**Default Value**: none

**mappings**

Provides prompts that enable you to delete any of the static address mappings.

**Valid Values**: An index number to identify the configured mapping. This number is displayed when you enter the **list mappings** command.

**Default Value**: none

## Disable

Use the **disable** command to disable NAT. You can disable NAT so that it will drop packets requiring translation or you can disable NAT so that it will pass packets requiring translation.

**Syntax:**

**disable nat**

drop

pass

**drop** Disables NAT so that it drops packets requiring translation.

**pass** Disables NAT so that it passes packets requiring translation.

## Enable

Use the **enable** command to enable NAT. Enabling NAT makes it ready to run, but it will not run until you use the **reset** command or restart the router.

**Syntax:**

**enable nat**

## List

Use the **list** command to list the public IP address reserve pools, the private IP address translate ranges, the mappings, the global settings, or all the NAT information.

**Syntax:**

**list** reserve

addresses

pools

translate

mappings

global

all

In the following example, times are displayed as hours, minutes, and seconds. Entry age is the time elapsed since the entry was last used. A binding means that traffic is flowing between these two addresses. The timeouts determine how much time will elapse after the last communication before a binding is dropped. See the **set** command for more information about timeouts.

**Example:**

```
NAT config>list all
NAT Globals:
NAT is ENABLED
Tcp Timeout....:   24:00:00
Non-Tcp Timeout:    0:01:00
NAT Reserved Address Pool(s):
Index First Address    Mask             Count NAPT Address     Pool Name
1     9.8.7.1          255.255.255.0    3     0.0.0.0          pool1
2     9.8.7.6          255.255.255.0    12    9.8.7.9          pool1
NAT Translate Range(s):
Index IP Address       IP Mask          Associated Pool Name
1     7.1.1.0          255.255.255.0    pool1
2     10.0.0.0         255.0.0.0        pool1
NAT Static Mapping(s):
Index Private Address:Port    Public Address.:Port
1     10.1.2.3        0       9.8.7.1        0
2     7.1.1.1         21      9.8.7.9        21
```

# Map

Use the **map** command to statically bind a host or server in the private network to a public address. This command, which can be used to set up servers in the private network, establishes an association at NAT startup that never changes.

Static mappings with the public and private port number 0 are NAT mappings; those with other values for the port numbers are NAPT mappings.

**Syntax:**

**map** *private-address private-port-number public-address public-port-number*

**private-address**

The private address of the workstation.

**Valid Values**: an Internet host address in valid IP format. This should be the address assigned to a station in the stub network that requires permanent access from the public network, such as a server.

**Default Value**: none

**private-port-number**

The TCP/UDP port number of the application running in the device with the private address. Entering **0** creates a NAT binding and entering another value creates a NAPT binding. Common port values for NAPT are 23 for Telnet, 21 for FTP, and 80 for HTTP.

**Valid Values**: 0 - 65535

**Default Value**: 0

**public-address**

The public IP address to which this private address is to be mapped. This must be a NAPT address for a NAPT mapping and a NAT address for a NAT mapping.

**Valid Values**: a valid IP address unique to the public network. The public network can be the Internet or an intranet, depending upon the design of the network.

**Default Value**: none

**public-port-number**
The port number of the packets to be translated at the public address. The value 0 represents all ports. Common values are 23 for Telnet, 21 for FTP, and 80 for HTTP.

**Valid Values**: 0 - 65535

**Default Value**: 0

In this example, the server with private IP address 10.11.12.200 accepts all traffic from the Internet; the server with private address 10.11.12.199 is a Telnet server and an FTP server.

**Example:**

```
map 10.11.12.200 0 9.8.7.2 0
map 10.11.12.199 23 9.8.7.9 23
map 10.11.12.199 21 9.8.7.9 21
```

## Reserve

Use the **reserve** command to create and append a range of IP addresses to a public address pool.

**Syntax:**

**reserve** *public-address mask number-in-group name napt-address*

**public-address**
The first public IP address in the sequence of addresses that make up this range or group in the pool. For example, if this group in the pool includes the 12 addresses in sequence from 9.8.7.6 through 9.8.7.17, this value is 9.8.7.6.

**Note:** To add another range of addresses to the public address pool, use the **reserve** command separately for each group, relating one group to another by using the same pool name. For example, addresses 9.8.7.6 through 9.8.7.17 can be configured in one group within pool1 and addresses 9.8.7.1 through 9.8.7.3 can be configured in another group within the same pool. Then, addresses 9.8.7.4 and 9.8.7.5 are not configured or used by that pool.

**Valid Values**: a valid IP address that is unique to the public network

**Default Value**: none

**mask** A mask to select bits from the IP address. The mask, like an Internet address, is 32 bits long. The 1s in the mask select the network or subnet part of the address. The 0s select the host portion. For example, the address 9.8.7.6 and the mask 255.255.0.0 includes the range of all addresses of which the first two bytes are 9.8 (that is, 9.8.0.0 through 9.8.255.255).

**Valid Values**: any valid IP mask

**Default Value**: none

**number-in-group**
Specifies how many sequential addresses, beginning with the *public-address*, are included in the group. For the addresses 9.8.7.6 through 9.8.7.17, this value is 12.

**Valid Values**: 1 - the value that can be defined by the IP mask

**Default Value**: none

**name** The name of the public address reserve pool. This string has to match the pool name on the corresponding **translate** command.

**Valid Values**: any name, using up to 16 printable characters; leading and trailing blanks are ignored.

**Default Value**: none

**napt-address**
The one IP address from the public address pool that will be used by Network Address Port Translation (NAPT). This address is used for TCP and UDP traffic to map multiple private addresses to the one NAPT address according to the protocol port number. Using NAPT is optional. If it is used, there can be only one NAPT address per public address pool. If there is no NAPT address for a pool or group, enter the value **0.0.0.0**. You need only enter the NAPT address once for the pool.

**Valid Values**: one of the public IP addresses. It does not necessarily have to be included in the range of values defined in the public address pool, but it must be in the same subnet.

**Default Value**: 0.0.0.0 (meaning no NAPT)

**Example:**

```
reserve 9.8.7.1 255.255.255.0 3  pool1 0.0.0.0
reserve 9.8.7.6 255.255.255.0 12 pool1 9.8.7.9
```

## Reset

Use the **reset** command to reset NAT. This command deletes all bindings, frees all memory used by NAT, and restarts NAT based on the current Talk 6 configuration. Resetting NAT does not disrupt any other components of the 2216.

**Syntax:**

**reset nat**

Note that if NAT encounters an invalid configuration, you will see a message to that effect. Review the NAT ELS messages to see why NAT initialization failed.

## Set

Use the **set** command to set TCP and non-TCP timeouts.

**Syntax:**

**set** **tcp**

**nontcp**

**tcp** *timeout*
The time that NAT maintains a TCP binding after the last message passes between the two bound workstations. A binding is the maintenance of the relationship between a private address and one of the public IP addresses.

**Valid Values**: 0 - 65535 minutes (0 minutes to about 45 days)

**Default Value**: 1440 minutes (24 hours)

**nontcp** *timeout*
The time that NAT maintains a binding that is not TCP after the last message passes between the two bound stations. A binding is the maintenance of the relationship between a private address and one of the public IP addresses.

**Valid Values**: 0 - 65535 minutes (0 minutes to about 45 days)

**Default Value**: 1 minute

## Translate

Use the **translate** command to add a subnet to the list of addresses that NAT will translate. Each subnet is a translate range. This command must be entered once for each translate range that NAT must know. Any number of translate ranges can use a single public address reserve pool.

**Syntax:**

**translate** *private-address mask name*

**private-address**
Any IP host or subnet address that should be translated.

**Valid Values**: an address in valid dotted decimal IP format. When ANDed with its subnet mask, this address identifies all addresses in a stub subnet. A stub subnet is a network that accesses the public network only through the router.

**Default Value**: none

**mask** **Valid Values**: The network or subnet mask associated with the stub network to be translated.

**Default Value**: class mask of the private address

**name** The name of the public address pool NAT should use for this range of private addresses.

**Valid Values**: any name, using up to 16 printable characters. It must match a public address pool name created by the **reserve** command.

**Default Value**: none

# Accessing the Network Address Translation Monitoring Environment

To access the NAT monitoring environment, type

```
* t 5
```

Then, enter the following command at the **+** prompt:

```
+ feature NAT
NAT>
```

The NAT> prompt appears.

# Network Address Translation Monitoring Commands

This section describes the IP Security monitoring commands. Enter these commands at the NAT> prompt.

*Table 136. NAT Monitoring Commands*

| Command | Function |
|---|---|
| ? (Help) | Displays all the commands available for this command level or lists the options for specific commands (if available). See “Getting Help” on page 10. |
| List | Lists information about NAT. |
| Reset | Causes the router to read in the NAT configuration and run according to the NAT access rules that have been configured. NAT does not affect the running of the router until you enter the **reset NAT** command. |
| Exit | Returns you to the previous command level. See “Exiting a Lower Level Environment” on page 11. |

## List

Use the **list** command to display information about the NAT configuration.

**Syntax:**

**list**
- all
- binding
- fragment
- global
- reserve
  - pools
  - addresses
- statistics
- translate

In the following example, times are displayed as hours, minutes, and seconds. Entry age is the time elapsed since the entry was last used. A binding means that a session is established between these two addresses. The timeouts determine how much time will elapse after the last communication before a binding is dropped. See the **set** command in Talk 6 for more information about timeouts.

**Example:**

```
NAT>list all
NAT Globals:
Current State    Tcp Timeout    Non-Tcp Timeout    Memory Usage (in bytes)
ENABLED             24:00:00            0:01:00           408

NAT Statistics:
Requests :     Passes      Drops      Holds
       0 :          0          0          0

NAT Address Binding(s):
Private Address//Port  Public Address//Port  Bind Type  Entry Age
        7.1.1.1    21         9.1.1.1    21  STATIC       0:00:13
```

```
       10.1.2.3     0         9.1.1.2     0  STATIC       0:00:13

NAT TCP Session Information:
Private Address//Port  Public Address//Port  Tcp State  Data Delta  Entry Age
        7.1.1.1    21         9.1.1.1    21  ESTAB'ED      0        0:00:56

NAT Translate Range(s):
Base Ip Address        Range Mask      Associated Reserve Pool
       7.1.1.0         255.255.255.0   carol
      10.0.0.0         255.0.0.0       carol

NAT Reserve Pool(s):
Reserve Pool     Pool Size    NAPT Address   1st Available Address
carol                21            9.1.1.1          9.1.1.12
------------------------------------------------------------
Number of Reserve Pools using NAPT.....:     1
Number of configured Reserved Addresses:    21

NAT Fragment Information:
Number of Entries    Number of Saved Fragments
          0                          0
```

## Reset

Use the **reset** command to reset NAT. This command deletes all bindings, frees all memory used by NAT, and restarts NAT based on the current Talk 6 configuration. Resetting NAT does not disrupt any other components of the 2216.

**Syntax:**

**reset nat**

# Appendix A. Quick Configuration Reference

**Important**

If you are attempting to configure or monitor your IBM 2216 and your service terminal is unreadable, see “Service Terminal Display Unreadable” in IBM 2216 Nways Multiaccess Connector Service and Maintenance Manual.

## Quick Configuration Tips

Before starting the Quick Configuration process, read these notes:

1. Attach an ASCII terminal to the service port to run the Quick Configuration program. See the *Installation and Initial Configuration Guide*.
2. Any existing configuration for a particular item will be removed if that item is configured through Quick Configuration.
3. Configuration is done at the level of the *interface*, which corresponds to a single *port* on an adapter. Because different types of adapters have differing numbers of ports, you may have to configure up to eight ports to activate all of the interfaces on the adapter (for example, the X.21 adapter, FC 2291).

   **Note:** The ESCON adapter or PCA can have up to 16 virtual interfaces configured on one physical interface. All of the virtual interfaces are associated with a single port.
4. Using the **add device** command, you must “add” all desired network interfaces or virtual interfaces for the adapters installed in your IBM 2216. This must be done prior to running Quick Configuration. To add an interface, see “Add” on page 68 .
5. Using the **network** command, you must enter the network interface configuration information. See “Network” on page 89.

## Making Selections

On the panels that you view when using the Quick Configuration program, the information shown in brackets, [ ], is the default. For example:

```
Configure Bridging? (Yes, No, Quit): [Yes]
```

- To use the default `Yes`, press **Enter**.
- To use a value other than the default, such as `No` or `Quit`, choose from the values in the parentheses.
- If no value appears in the brackets, there is no default and you must type a value.

## Exiting and Restarting

- To restart the current Quick Configuration section at any time, type **r**. For example, if you are in the Interface Configuration section, type **r** and press **Enter** to return to the beginning of that section.
- To exit Quick Configuration, type **q** and press **Enter**. The `Config>` prompt will appear.
- To restart Quick Configuration from the `Config>` prompt, type **qc** and press **Enter**.

## When You're Done

- Once you have completed your configuration, you must restart the IBM 2216 for the configuration to take effect. At the end of the Quick Configuration program, you are given this option.

# Starting the Quick Configuration Program

The following sections describe sample configurations using the Quick Configuration program (**qconfig**).

To start the quick configuration program, enter **qc** at the `Config>` prompt.

The program displays the following panel after starting.

```
Router Quick Configuration for the following:

o   Bridging
        Spanning Tree Bridge (STB)
        Source Routing Bridge (SRB)
        Source Routing Transparent Bridge (SRT)
o   Protocols
        IP (including OSPF, RIP, and SNMP)
        IPX
        DNA (DECnet)

Event Logging will be enabled for all configured subsystems
with logging level 'Standard'

Note: Please be warned that any existing configuration for a particular item
will be removed if that item is configured through Quick Configuration
```

*Event logging* records system activity, status changes, data transmission and reception, data and internal errors, and service requests. The logging level is set to standard (the default). For more information about error logging, refer to the *Event Logging System Messages Guide*.

During Quick Configuration you can:

1. Configure bridging
2. Configure protocols
3. Restart the router

# Configuring LAN Emulation

If you added an ATM device, you will see the following prompts:

```
************************************************************
LAN Emulation Configuration
************************************************************

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

Configure LAN Emulation? (Yes, No, Quit): [Yes]
```

You can configure either Token-Ring or Ethernet LAN Emulation clients from this screen.

## Configuring Bridging

```
*********************************************
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]
```

1. In response to `Configure Bridging`, take one of the following actions:
   - Enter **y** to display the bridging configuration prompts. The prompts that appear depend on your network configuration.
   - Enter **n** to skip the bridging configuration and continue with quick configuration.
   - Enter **q** to exit quick configuration. This displays the `Config>` prompt. To reenter quick configuration, enter **qc** after this prompt.
2. If you choose to configure bridging, Spanning Tree Bridging (STB) will be enabled on all LAN interfaces. You will see the following panels:

   ```
   Type 'r' any time at this level to restart Bridging Configuration

   STB will be enabled on all LAN interfaces
   ```

   Enter **y** to configure SRT bridging. Otherwise, enter **n**. For each Token-Ring interface in the configuration, you will be prompted to enable Source Routing on the interface.

   ```
   Configure SRT Bridging? (Yes, No): [Yes]
   You are now configuring the Source Routing part of SRT Bridging
   Bridge Number (hex) of this Router (1-F): [A]
   ```

3. Enter a bridge number, which is a hexadecimal value from 1 to F that is unique between two parallel segments.

   ```
   Interface 0 (Port 1) is of type Token Ring
   Configure Source Routing on this interface (Yes, No): [Yes]
   ```

4. Enter **y** to configure source routing on the interface. The console displays the next two lines.

   ```
   Configuring Interface 0 (Port 1)
   Segment Number (hex) of this Interface (1-FFF): [A1]
   ```

   **Note:** The port number increases by one because source routing bridging does not allow a port number of zero.

   A unique hexadecimal value from 1 to FFF is assigned to each interface. The interfaces on each ring (segment) have the same segment number, but the segment number is unique to each ring.

   These prompts appear for each Token Ring interface.

```
Interface 1 (Port 2) is of type Token Ring
Configure Source Routing on this interface? (Yes, No): [Yes]
Configuring Interface 1 (Port 2)
Segment Number (hex) of this Interface (1-FFF): [A2]
```

If more than two interfaces are configured for source routing, enter a unique hexadecimal value from 1 to FFF unique for the internal virtual segment.

```
Virtual Segment Number (hex) of this Router (1-FFF): [A4]
```

5. A panel similar to the following is displayed:

```
This is all configured bridging information:

       Interfaces configured for STB:

       Interface #    Port #     Interface Type

           0          1          Token Ring
           1          2          Token Ring

The Source Routing part of SRT Bridging has been enabled

Bridge Number of this Router: A

Interfaces configured for Source Routing:

       Interface #     Port#      Segment #   Interface Type

           0             1          A1        Token Ring
           1             2          A2        Token Ring

Virtual Segment Number of this Router:  A4

Save this Configuration? (Yes, No): [Yes]
```

6. Enter **y** to save the bridging configuration and continue with quick configuration. Enter **n** to re-display the bridging configuration prompts.

   If you enter **y**, the following message appears:

```
Bridging configuration saved
```

## Configuring Protocols

After you save the bridging configuration, you will see the following panel:

```
*******************************************
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]
```

Take one of the following actions:

- Enter **y** to configure the protocols.
- Enter **n** to skip protocol configuration and continue with quick configuration.
- Enter **q** to exit quick configuration.

You will first configure IP, then IPX, and then DECnet.

## Configuring IP

When you answer **y** to the Configure Protocol panel, quick configuration displays the following messages:

```
Type 'r' any time at this level to restart Protocol configuration

Configure IP? (Yes, No): [Yes]
```

1. Take one of the following actions:
   - Enter **y** to configure IP.
   - Enter **n** to skip IP configuration and continue with quick configuration.

The following lines appear for each interface.

```
Configuring Per-Interface IP Information

Configuring Interface 0 (Token Ring)
Configure IP on this interface? (Yes, No): [Yes]
IP Address: [ ] 128.185.141.1
Address Mask: [255.255.0.0]
```

2. Enter the IP address in decimal notation for example, 128.185.142.20. The console displays one of the following error messages if you enter an invalid IP address:

   ```
   Bad address, please try again.

   This address has already been assigned.  Enter a different address
   ```

   Address mask is a decimal value that reflects the IP network or subnetwork to which this interface is attached.

   For more information about IP addressing or address masks, refer to the *Protocol Configuration and Monitoring Reference*, or consult your network administrator.

   ```
   Per-Interface IP Configuration complete

   Configuring IP Routing Information
   Enable Dynamic Routing (Yes, No): [Yes]
   ```

3. Enter **y** if you want the routing protocols (RIP or OSPF) to build the routing tables. Enter **n** to manually add IP address destinations to the routing tables (static routes).

   ```
   Enable OSPF? (Yes, No): [Yes]
   ```

4. Enter **y** to enable the OSPF routing protocol as the primary dynamic IP routing protocol. RIP will be enabled only to send advertisements, not to receive them. Enter **n** if you do not want to use OSPF. RIP will be enabled to send and receive advertisements.

   ```
   OSPF Enabled with Max routes = 1000  and Max routers = 50
   ```

   Max routes is the maximum number of autonomous system (AS) external routes imported into the OSPF routing domain. Max routers is the maximum number of OSPF routers in the routing domain.

```
Routing Configuration Complete

SNMP will be configured with the following parameters:

Community: public
Access:    READONLY

If you plan to use the graphical configuration tool
to download a configuration, it requires the definition
of a community name with read_write_trap access.

Define community with read_write_trap access ? (Yes, No):  [Yes]


This is the information you have entered:

          Interface #        IP Address     Address Mask

             0               128.185.141.1  255.255.255.0
             1               128.185.142.1  255.255.255.0
             2               128.185.143.1  255.255.255.0

OSPF is configured, and RIP is configured only for 'sending'

SNMP has been configured with the following parameters:

     Community: public
     Access:    read_trap

     Community: dana
     Access:    read_write_trap

Save this configuration? (Yes, No): [Yes]
```

5. Enter **y** to save the IP configuration and continue with quick configuration. Enter **n** to re-display the protocol configuration prompts.

## Configuring IPX

After you save the IP configuration, you will see the following messages:

```
Configure IPX? (Yes, No): [Yes]
```

1. Enter **y** to configure IPX. Enter **n** to skip IPX configuration and continue with quick configuration.

   You will see messages similar to the following:

   ```
   Type 'r' any time at this level to restart IPX Configuration
   IPX Configuration is already present
   Configure IPX anyway? (Yes, No): [No] yes
   ```

2. Enter **y** to replace the existing configuration. Enter **n** to keep the current configuration and continue.

   ```
   Configuring Per-Interface IPX Information

   Configuring Interface 0 (Token Ring)
   Configure IPX on this interface? (Yes, No): [Yes]
   ```

3. The next messages and your responses depend on whether you are configuring Token-Ring, FDDI, or Ethernet.

   **Configuring IPX for Token-Ring:**

   a. The following prompt is displayed:

```
Token Ring encapsulation (frame) type? (TOKEN–RING MSB, TOKEN–RING LSB,
    TOKEN–RING_SNAP MSB, TOKEN–RING_SNAP LSB): [TOKEN–RING MSB]
```

b. Enter the encapsulation type used by the IPX protocol on your Token-Ring end stations.

`Token–Ring MSB:` Most common encapsulation type and the default. The IBM 2216 builds outgoing packets with a 3-byte 802.2 header, (0xE0, 0xE0, 0x03). It sends the source and destination addresses in MSB (most significant bit), or noncanonical, format, which is the native address format for Token-Ring.

`Token–Ring LSB` Same as Token-Ring MSB except the IBM 2216 sends the addresses in LSB (least significant bit), or canonical, format.

`Token-Ring SNAP MSB` The IBM 2216 builds outgoing packets with an 8-byte 802.2/SNAP header (0xAA, 0xAA, 0x03, 0x00, 0x00, 0x00, 0x81, 0x37). It sends the source and destination addresses in most significant bit (MSB), or noncanonical, format.

`Token-Ring SNAP LSB` Same as Token-Ring SNAP MSB except the IBM 2216 sends the addresses in LSB, or canonical, format.

**Configuring IPX for Ethernet:**

a. The following prompts are displayed:

```
Ethernet encapsulation type? (ETHERNET_8022, ETHERNET_8023, ETHERNET_ii,
ETHERNET_SNAP): [ETHERNET_8023]
```

b. Enter the encapsulation type used by the IPX protocol on your Ethernet end stations.

`Ethernet_8022` Packet includes an 802.2 header.

`Ethernet_8023` Uses an IEEE 802.3 packet format without the 802.2 header. This is the default and the default for NetWare versions prior to 4.0. Ethernet 802.3 does not conform to the IEEE 802 standards because it does not include an 802.2 header. It may cause problems with other nodes on the network.

`Ethernet_II` Uses Ethernet type 8137 as the packet format. This format is required if you are using NetWare VMS on the Ethernet. This is the default for NetWare Versions 4.0 and higher.

`Ethernet_SNAP` Uses the 802.2 format with a SNAP header. This encapsulation type is meant to be compatible with token-ring SNAP encapsulation. However, it violates IEEE standards and is not interoperable across conformal bridges.

**Configuring IPX for FDDI:**

a. The following prompts are displayed:

```
FDDI encapsulation (frame) type? (FDDI, FDDI_SNAP): [FDDI_SNAP]
```

b. Enter the encapsulation type used by the IPX Protocol for your FDDI end stations.

| | |
|---|---|
| fddi | Sets the encapsulation type to FDDI IEEE 802.2. |
| fddi_snap | Sets the encapsulation type to FDDI SNAP. |

```
Network Number (hex) (1-FFFFFFFD):[1] 1
```

4. Assign an IPX network number to the associated directly connected network. Every IPX interface must have a unique network number.

```
Configuring Interface 1 (WAN PPP)
Configure IPX on this interface? (Yes, No): [Yes]
Network Number (hex) (1-FFFFFFFD): [1] 2

Enable IPXWAN? (Yes, No): [No] yes

Configuring Interface 2 (WAN PPP)
Configure IPX on this interface? (Yes, No): [Yes]
Network Number (hex) (1-FFFFFFFD):[1] 3

Enable IPXWAN? (Yes, No): [No] yes

Host Number for Serial Lines: (000000000000) 1

Configure IPXWAN NodeID? (Yes, No): [Yes]
NodeID (hex) (1 - FFFFFFFD): [1] 4
```

   If enabled, the IPXWAN protocol negotiates routing parameters to be used on the PPP serial interface before IPX packet forwarding begins. IPXWAN is not required to forward IPX packets on PPP serial interfaces. The IPXWAN Node ID is a unique IPX network number that identifies the router, and is required if IPXWAN is enabled on any network interfaces.

5. Host number is a unique 12-digit hexadecimal value assigned to an IPX router. It is required because serial lines do not have hardware node addresses from which to build a host number.

```
This is the information you have entered:

               Per-Interface Configuration Information

Ifc   IPX Net (hex)   Encapsulation       IPXWAN

0     1               TOKEN-RING MSB      Not Configured
1     2                                   Enabled
2     3                                   Enabled

  Host Number for Serial Lines: 000000000001
  IPXWAN Node ID = 4
  IPX Router Name = ipx_router-4
Save this configuration? (Yes, No): [Yes]
```

6. Enter **y** to save the IPX configuration and continue with quick configuration. Enter **n** to re-display the IPX configuration prompts.

   If you enter **y**, the following message appears:

```
IPX configuration saved
```

## Configuring DECnet (DNA)

After you save the IPX configuration, you will see the following messages.

```
IPX Configuration saved

Configure DNA? (Yes, No): [Yes]
```

1. Enter **y** to configure DNA. Enter **n** to skip DNA configuration and continue with quick configuration.

```
Type 'r' any time at this level to restart DNA Configuration

Configuring Global DNA information

Highest Node Number (decimal) (1-1023): [32]
Router Level (Level1, Level2, DEC Level1, DEC Level2):
  [ Level2]
Highest Area (decimal) (1-63): [63]
Node Address (area.node): (63.32)
```

The above configuration fields are configured with the following considerations:

**Highest Node Number**
Is the highest node address in the router's area. Setting it excessively high will affect the routers efficiency and require excess storage.

**Router Level**
Identifies whether the router is a Level 1 or Level 2 router. A Level 1 router keeps track of all nodes in its area and does not care about nodes outside its area. A Level 2 router routes traffic between areas.

Normally you should select Level1 or Level2 with the following exception: select DEC Level1 or DEC Level2 only when this router must communicate over X.25 networks with routers conforming to the DEC X.25 standard.

**Highest Area**
This number should be at least as high as the highest area number in the overall network.

**Node Address**
Is the node ID of this router and must be unique in the network.

When you press Enter, the following is displayed:

```
Configuring Per-Interface DNA Information

Configuring Max Routers on each interface

Configuring Interface 0 (Ethernet)
Configure DNA on this interface? (Yes, No) [YES]
Max Routers (decimal) (1-33): [16]

Configuring Interface 1 (WAN PPP)
Configure DNA on this interface? (Yes, No) [Yes]

Configuring Interface 2 (Token Ring)
Configure DNA on this interface? (Yes, No) [Yes]
Max Routers (decimal) (1-33): [16]
```

2. Enter **y** for every interface that will be connected to the DECnet network. For LANs, Max Routers specifies how many other routers may be on this circuit. For router efficiency and memory requirements set this argument to a few more than the total number of adjacent routers on this circuit.

   The following panel is displayed:

```
This is the information you have entered:

        Global Configuration Information

           Highest Node Number:       32
           Router Level:              Level2
           Highest Area:              63
           Node Address:              63.32

        Pre-Interface Configuration Information
      Interface Number           Max Routers

              0                          16
              1                          1
              2                          16

Save this configuration? (Yes, No): [Yes]
```

3. Enter **y** to save the DECnet configuration and continue with the quick configuration. Enter **n** to re-display the DECnet configuration prompts.

   If you enter **y**, the following message appears:

```
DNA Configuration Saved
```

## Restarting the IBM 2216

After configuring the protocols, you will receive the following message:

```
Quick Config Done
Do you want to write this configuration? (Yes, No): [Yes]
```

Enter **y** to save your changes and display the following information:

```
Default config file written successfully.

Configuration was written.
The system must be restarted for this configuration to take effect.
```

Enter **reload** at the OPCON prompt (*) to restart the IBM 2216 with the new configuration. To change or view the current configuration, enter **qc**.

# Appendix B. 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 C. 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 of 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 of 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 D. Licensed Program Materials Availability

This program, 5765-C90, 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.

# Appendix E. Remote AAA Attributes

This section contains the remote AAA Attributes use by Radius, TACACS and TACACS+ servers.

## Radius

IBM Vendor ID: 211

**Authorization Attributes**

**Standard Drafted**

| | |
|---|---|
| TUNNEL_TYPE | 64 |
| TUNNEL_MEDIUM_TYPE | 65 |
| TUNNEL_CLIEN_TYPE | 66 |
| TUNNEL_SERVER_EP | 67 |
| TUNNEL_CONN_ID | 68 |
| TUNNEL_PASSWORD | 69 |

values

| | | |
|---|---|---|
| TUNNEL_TYPE | | integer |
| 3 | L2TP | |
| TUNNEL_MEDIUM_TYPE | | integer |
| 1 | IP | |
| TUNNEL_SERVER_EP | | string |
| | ip address | |

**IBM Vendor Specific**

| | |
|---|---|
| NAS_TUNNEL_PASSWORD | 101 |
| CALLBACK_FLAGS | 210 |
| ENCRYPTION | 211 |
| HOSTNAME | 213 |
| DIALOUT | 214 |
| SUBNETMASK | 215 |
| PRIVILEGE | 216 |

## Keywords

Keywords are used for Radius servers that allow the entry of vendor specific fields `<keyword>=<value>`.

| | |
|---|---|
| KWD_CALLBACK_FLAGS | CBF |
| KWD_ENCRYPTION | ENC |
| KWD_HOSTNAME | HSN |
| KWD_DIALOUT | DOF |
| KWD_SUBNETMASK | SNM |

| | |
|---|---|
| KWD_PRIVELGE | PRV |

Values

PRIVILEGE:

| | |
|---|---|
| ADMIN | |
| OPER | |
| MONITOR | |

CALLBACKFLAGS

| | |
|---|---|
| REQ | required callback |
| ROAM | roaming callback |

DIALOUT

| | |
|---|---|
| TRUE | enable dialout for this user |
| FALSE | disable dialout for this user |
| ONLY | only allow dialout for this user (not dial in) |

## TACACS+

**Authentication**

**Authorization**

PPP service=ppp protocol=ip
LOGIN service=shell cmd=null pri_lvl*0

Standard TACACS+ Attributes

- service
- protocol
- cmd
- addr
- timeout
- priv_lvl
- callback-dialstring

IBM Specific Attributes

| | |
|---|---|
| encryption_key | 16 hex characters |
| dial_out | TRUE FALSE ONLY |

**Accounting**

- task_id
- start_time
- stop_time
- elasped_time
- timezone
- event
- reason
- bytes
- bytes_in
- bytes_out
- paks

paks_in
paks_out
status
err_msg

# List of Abbreviations

**AARP** AppleTalk Address Resolution Protocol

**ABR** area border router

**ack** acknowledgment

**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** ATM real interface

**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

**AVI** ATM virtual interface

**ayt** are you there

**BAN** Boundary Access Node

**BBCM** Bridging Broadcast Manager

**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

**CIP** Classical IP

**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

**DCS** Directly connected server

**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

**DSU** data service unit

**DTE** data terminal equipment

**DTR** data terminal ready

**Dtype** destination type

**DVMRP**
Distance Vector Multicast Routing Protocol

**E1** 2.048 Mbps transmission rate

**EDEL** end delimiter

**EDI** error detected indicator

**EGP** Exterior Gateway Protocol

**EIA** Electronics Industries Association

**ELAN** Emulated LAN

**ELAP** EtherTalk Link Access Protocol

**ELS** Event Logging System

**ESI** End system identifier

**EST** Eastern Standard Time

**Eth** Ethernet

**fa-ga** functional address-group address

**FCS** frame check sequence

**FECN** forward explicit congestion notification

**FIFO** first in, first out

**FLT** filter library

**FR** Frame Relay

**FRL** Frame Relay

**FTP** File Transfer Protocol

**GMT** Greenwich Mean Time

**GOSIP**
Government Open Systems Interconnection Profile

**GTE** General Telephone Company

**GWCON**
Gateway Console

**HDLC** high-level data link control

**HEX** hexadecimal

**HPR** high-performance routing

**HST** TCP/IP host services

**HTF** host table format

**IBD** Integrated Boot Device

**ICMP** Internet Control Message Protocol

**ICP** Internet Control Protocol

**ID** identification

**IDP** Initial Domain Part

**IDP** Internet Datagram Protocol

**IEEE** Institute of Electrical and Electronics Engineers

**Ifc#** interface number

**IGP** interior gateway protocol

**InARP** Inverse Address Resolution Protocol

**IP** Internet Protocol

**IPCP** IP Control Protocol

**IPPN** IP Protocol Network

**IPX** Internetwork Packet Exchange

**IPXCP** IPX Control Protocol

**ISDN** integrated services digital network

**ISO** International Organization for Standardization

**Kbps** kilobits per second

**LAC** L2TP Network Access Concentrator

**LAN** local area network

**LAPB** link access protocol-balanced

**LAT** local area transport

**LCS** LAN Channel Station

**LCP** Link Control Protocol

**LED** light-emitting diode

**LF** largest frame; line feed

**LIS** Logical IP subnet

**LLC** logical link control

**LLC2** logical link control 2

**LMI** local management interface

**LNS** L2TP Network Server

**LRM** LAN reporting mechanism

**LS** link state

**LSA** link state advertisement

**LSA** Link Services Architecture

**LSB** least significant bit

**LSI** LAN shortcuts interface

**LSreq** link state request

**LSrxl** link state retransmission list

**LU** logical unit

**MAC** medium access control

**Mb** megabit

**MB** megabyte

**Mbps** megabits per second

**MBps** megabytes per second

**MC** multicast

**MCF** MAC filtering

**MIB** Management Information Base

**MIB II** Management Information Base II

**MILNET**
military network

**MOS** Micro Operating System

**MOSDDT**
Micro Operating System Dynamic Debugging Tool

**MOSPF**
Open Shortest Path First with multicast extensions

**MPC** Multi-Path Channel

**MPC+** High performance data transfer (HPDT) Multi-Path Channel

**MSB** most significant bit

**MSDU** MAC service data unit

**MRU** maximum receive unit

**MTU** maximum transmission unit

**nak** not acknowledged

**NAS** Nways Switch Administration station

**NBMA** Non-Broadcast Multiple Access

**NBP** Name Binding Protocol

**NBR** neighbor

**NCP** Network Control Protocol

**NCP** Network Core Protocol

**NDPS** non-disruptive path switching

**NetBIOS**
Network Basic Input/Output System

**NHRP** Next Hop Resolution Protocol

**NIST** National Institute of Standards and Technology

**NPDU** Network Protocol Data Unit

**NRZ** non-return-to-zero

**NRZI** non-return-to-zero inverted

**NSAP** Network Service Access Point

**NSF** National Science Foundation

**NSFNET**
National Science Foundation NETwork

**NVCNFG**
nonvolatile configuration

**OPCON**
Operator Console

**OSI** open systems interconnection

**OSICP**
OSI Control Protocol

**OSPF** Open Shortest Path First

**OUI** organization unique identifier

**PC** personal computer

**PCA** parallel channel adapter

**PCR** peak cell rate

**PDN** public data network

**PING** Packet internet groper

**PDU** protocol data unit

**PID** process identification

**P-P** Point-to-Point

**PPP** Point-to-Point Protocol

**PROM** programmable read-only memory

**PU** physical unit

**PVC** permanent virtual circuit

**RAM** random access memory

**RD** route descriptor

**REM** ring error monitor

**REV** receive

**RFC** Request for Comments

**RI** ring indicator; routing information

**RIF** routing information field

**RII** routing information indicator

**RIP** Routing Information Protocol

**RISC** reduced instruction-set computer

**RNR** receive not ready

**ROM** read-only memory

**ROpcon**
Remote Operator Console

**RPS** ring parameter server

**RTMP** Routing Table Maintenance Protocol

**RTP** RouTing update Protocol

**RTS** request to send

**Rtype** route type

**rxmits** retransmissions

**rxmt** retransmit

**s** second

**SAF** source address filtering

**SAP** service access point

**SAP** Service Advertising Protocol

**SCR** Sustained cell rate

**SCSP** Server Cache Synchronization Protocol

**sdel** start delimiter

**SDLC** SDLC relay, synchronous data link control

**seqno** sequence number

**SGID** sever group id

**SGMP** Simple Gateway Monitoring Protocol

**SL** serial line

**SMP** standby monitor present

**SMTP** Simple Mail Transfer Protocol

**SNA** Systems Network Architecture

**SNAP** Subnetwork Access Protocol

**SNMP** Simple Network Management Protocol

**SNPA** subnetwork point of attachment

**SPF** OSPF intra-area route

**SPE1** OSPF external route type 1

**SPE2** OSPF external route type 2

**SPIA** OSPF inter-area route type

**SPID** service profile ID

**SPX** Sequenced Packet Exchange

**SQE** signal quality error

**SRAM** static random access memory

**SRB** source routing bridge

**SRF** specifically routed frame

**SRLY** SDLC relay

**SRT** source routing transparent

**SR-TB** source routing-transparent bridge

**STA** static

**STB** spanning tree bridge

**STE** spanning tree explorer

**STP** shielded twisted pair; spanning tree protocol

**SVC** switched virtual circuit

**TB** transparent bridge

**TCN** topology change notification

**TCP** Transmission Control Protocol

**TCP/IP** Transmission Control Protocol/Internet Protocol

**TEI** terminal point identifier

**TFTP** Trivial File Transfer Protocol

**TKR** token ring

**TMO** timeout

**TOS** type of service

**TSF** transparent spanning frames

**TTL** time to live

**TTY** teletypewriter

**TX** transmit

**UA** unnumbered acknowledgment

**UDP** User Datagram Protocol

**UI** unnumbered information

**UTP** unshielded twisted pair

**VCC** Virtual Channel Connection

**VINES** VIrtual NEtworking System

**VIR** variable information rate

**VL** virtual link

**VNI** Virtual Network Interface

**VR** virtual route

**WAN** wide area network

**WRS** WAN restoral/reroute

**X.25** packet-switched networks

**X.251** X.25 physical layer

**X.252** X.25 frame layer

**X.253** X.25 packet layer

**XID** exchange identification

**XNS** Xerox Network Systems

**XSUM** checksum

**ZIP** AppleTalk Zone Information 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.** (1) Operational. (2) Pertaining to a node or device that is connected or is available for connection to another node or device.

**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.

**agent.** A system that assumes an agent role.

**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.** Asynchronous 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)

**CCITT.** International Telegraph and Telephone Consultative Committee. This was an organization of the International Telecommunication Union (ITU). On 1 March 1993 the ITU was reorganized, and responsibilities for standardization were placed in a subordinate organization named the Telecommunication Standardization Sector of the Telecommunication Union (ITU-TS). “CCITT” continues to be used for recommendations that were approved before the reorganization.

**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)*.

**channelization.** The process of breaking the bandwidth on a communication line into a number of channels, possibly of different size. Also called ***time division multiplexing*** (TDM).

**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 database (CDB).** A database that stores the configuration parameters of one or several devices. It is prepared and updated using the configuration program.

**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*.

**connection.** In data communication, an association established between functional units for conveying information. (I) (A)

**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)*.

**CU Logical Address.** The Control Unit address defined in the host for the 2216. This value is defined in the host Input/Output Configuration Program (IOCP) by the CUADD statement on the CNTLUNIT macro instruction. The Control Unit Address must be unique for each logical partition defined on the same host.

# 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)

**dependent LU requester (DLUR).** An APPN end node or an APPN network node that owns dependent LUs, but requests that a dependent LU server provide the SSCP services for those dependent LUs.

**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.

**device address.** The unit address transmitted on the channel path to select a 2216 device. It is also referred to as subchannel number in S/370 I/O architecture. This value is defined in the host IOCP by the UNITADD statement on the CNTLUNIT macro instruction for the real device.

**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.

**EIA unit.** A unit of measure, established by the Electronic Industries Association, equal to 44.45 millimeters (1.75 inches).

**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.

**equivalent capacity.** In the NBBS architecture, the minimum amount of bandwidth needed by a connection to ensure that the packet loss ratio is below a specified threshold.

**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

**fax.** Hardcopy received from a facsimile machine. Synonymous with ***telecopy.***

**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.

**flash memory.** A data storage device that is programmable, erasable, and does not require continuous power. The chief advantage of flash memory over other programmable and erasable data storage devices is that it can be reprogrammed without being removed from the circuit board.

**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.

**front-end processor.** A processor such as the IBM 3745 or 3174, that relieves a main frame from the communication control tasks.

# 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.

**hot pluggable.** Refers to an hardware component that can be installed or removed without disturbing the operation of any other resource that is not connected to, or dependant on, this component.

**hub (intelligent).** A wiring concentrator, such as the IBM 8260, that provides bridging and routing functions for LANs with different cables and protocols.

**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 Digital Network Exchange (IDNX).** A processor integrating voice, data, and image applications. It also manages the transmission resources, and connects to multiplexers and network management support systems. It allows integration of equipment from different vendors.

**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.

**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)*.

**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.

**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.

**IPPN.** The interface that other protocols can use to transport data over IP.

**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).

# J

**jitter.** (1) Short-term non-cumulative variations of the significant instants of a digital signal from their ideal positions in time. (2) Undesirable variations of a transmitted digital signal. (3) Variations in the network delay.

# 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 Address.** For the 2216 with an ESCON Channel Adapter, a port number determined as follows: If one ESCD is in the communication path, it is the ESCON Director (ESCD) port number that is attached to the host. If two ESCDs are in the path, it is the host-side port number of the ESCD defined with the dynamic connection. When no ESCD is in the communication path, this value must be set to X'01'.

**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 partition.** A number assigned to a partition in a host that can operate in logically partitioned (LPAR) mode. In LPAR mode, the ESCON adapter can share a physical fiber connection with multiple host partitions.

**Logically Partitioned (LPAR) mode.** A function of some host processors in which processing is divided into logical partitions (LPs) to provide the appearance of multiple processors. In LPAR mode, the ESCON adapter can share a physical fiber connection with multiple host partitions.

**LP.** logical partition

**LP number.** Logical partition number. This allows multiple logical host partitions, LPs, to share one ESCON fiber. This value is defined in the host Input/Output Configuration Program (IOCP) by the RESOURCE macro instruction. If the host is not using EMIF, use the default of 0 for the LP number.

**LPAR.** logically partitioned

**LPAR mode.** Logically Partitioned (LPAR) mode.

**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 access.** An Nways Switch that connects a network management station, or a change control server, to an NBBS network.

**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.** (1) MIB module. (2) Management Information Base.

**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.

**module.** In the Nways Switch, a packaged functional hardware unit containing logic cards, connectors, and lights. The modules are used to package adapters, line interface couplers, voice server extensions, and other components. All modules are ***hot pluggable*** in the logic subracks.

**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.

**multipath channel (MPC).** A channel protocol that uses multiple unidirectional subchannels for VTAM-to-VTAM bidirectional communication.

**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 Access Server (NAS).** A device providing temporary, on-demand network access to users. This access is point-to-point using PSTN or ISDN lines.

**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 control.** The functions of the NBBS architecture that are performed by a control point of the Nways Switch to:

- Allocate and control the Nways Switch resources
- Provide the topology and directory services
- Select the routes
- Control congestion

**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 (NMS).** A station that runs NetView/AIX and the Nways Switch Manager. It manages the NBBS network topology, accounting, performance, configuration updates, and problem analysis. A network management station is connected to its management access Nways Switch through an Ethernet LAN.

**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 Support Center.** A location from which IBM provides remote support to NBBS networks.

**network support station.** The processor used to locally operate and service the Nways Switch. It is used by the Nways Switch administrator or service personnel.

**network user address (NUA).** In X.25 communications, the X.121 address containing up to 15 binary code digits.

**Networking BroadBand Services (NBBS).** An IBM architecture for the high-speed networking that complements the ATM standards and provides the following functions:

- Access services

- Transport services
- Network control

**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*.

**Non-Return-to-Zero Changes-on-Ones Recording (NRZ-1).** A recording method in which the ones are represented by a change in the condition of magnetization, and zeros are represented by the absence of change. Only the one signals are explicitly recorded. (Previously called ***non-return-to-zero inverted***, NRZI, recording.)

**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.

**Nways Switch.** Synonymous with IBM 2220 Nways BroadBand Switch.

**Nways Switch configuration station.** A dedicated OS/2 station running a stand-alone version of the Nways Switch Configuration Tool (NCT). It is used to generate a network configuration database and should be installed as a remote console.

# 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 loss ratio.** The probability that a packet will not reach its destination or not reach it within a specified time.

**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 adapter.** A module, in models of the Nways Switch other than the 2216, running the code that provides the access services of the NBBS architecture to the port lines. In the 2216 the functions of the port adapter and the trunk adapter are combined in the Multiple Port/Trunk Adapter (MPTA).

**port line.** A communication line that connects an external user device to an Nways Switch and, thus, to the NBBS Network. It can have different access services and interfaces: circuit emulation service (CES), pulse code modulation (PCM), high-level data link control (HDLC), or frame relay (FR). In the Nways Switch, each port line is associated with one (or several) NBBS port(s).

**port number.** In Internet communications, the identification of an application entity to the transport service.

**potential connection.** In the NBBS architecture, a predefined connection between two devices external to the NBBS Network. It is defined by configuration parameters stored at one of the end-point Nways Switches.

**private branch exchange (PBX).** A private telephone exchange for transmission of calls to and from the public telephone network.

**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)

**pulse code modulation (PCM).** A standard adopted for the digitalization of an analog voice signal. In PCM, the voice is sampled at a rate of eight kHz and each sample is coded in an 8-bit frame. In an NBBS Network, PCM is an alternative to circuit emulation services (CES) to carry voice and fax data.

# Q

**quality of service (QoS).** In the NBBS architecture, the quality of service guarantees the characteristics of a network connection. It concerns mainly end-to-end delay, jitter, and packet loss ratio.

# 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.

**real-time processing.** The manipulation of data that are required, or generated, by some process while the process is in operation. Usually the results are used to influence the process, and perhaps related processes, while it is occurring.

**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 console.** A station running OS/2, TCP/IP, and the remote Nways Switch Resource Control program. It can be connected to any network support station to operate and service the Nways Switch remotely. The connection may be through:

- A switched line using a modem
- The NBBS Network, if the remote console is connected to its access Nways Switch through an Ethernet LAN.

Any network support station can be used as a remote console of another network support station.

**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.

**resource.** In the Nways Switch, an hardware element or a logical entity created by the Control Program. For example, the adapters, LICs, and lines are physical resources. The control points, NBBS trunks, NBBS ports, and connections are logical resources. In an NBBS Network, the resources must be configured before being operated.

**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

**SAP.** See service access point.

**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.

**Serial Line Internet Protocol (SLIP).** A protocol used over a point-to-point connection between two IP hosts over a serial line, for example, a serial cable or an RS232 connection into a modem, over a telephone line. In an NBBS network, the SLIP is used over a connection between a network support station and an IBM Network Support Center (NSC).

**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.** (1) An endpoint for communication between processes or application programs. (2) The abstraction provided by the University of California's Berkeley Software Distribution (commonly called Berkeley UNIX or BSD UNIX) that serves as an endpoint for communication between processes or applications.

**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)*.

**synchronous optical network (SONET).** A US standard for transmitting digital information over optical interfaces. It is closely related to the synchronous digital hierarchy (SDH) recommendation.

**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.** In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (I) (A)

**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

**TCP/IP.** (1) Transmission Control Protocol/Internet Protocol. (2) A UNIX-like/Ethernet-based system-interconnect protocol originally developed by the US Department of Defense. TCP/IP facilitated ARPANET (Advanced Research Projects Agency Network), a packet-switched research network for which layer 4 was TCP and layer 3, IP.

**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 division multiplexing (TDM).** See ***channelization***.

**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*.

**transport services.** The functions of the NBBS architecture that are performed by an MPTA of the Nways Switch to:

- Support the attachment of trunk lines to the Nways Switch
- Maximize the bandwidth utilization
- Guarantee the qualities of service
- Transfer packets between Nways Switches
- Manage logical queues and schedule transmission

**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.

**trunk adapter.** A module, in models of the Nways Switch other than the 2216, running the code that provides the transport services of the NBBS architecture to the trunk lines. In the 2216, the functions of the port adapter and the trunk adapter are combined in the Multiple Port/Trunk Adapter (MPTA).

**trunk line.** A high-speed line connecting two Nways Switches. It can be a coaxial cable, fiber cable, or radio wave, for example, and may be leased from telecommunication companies. In the Nways Switch, each trunk line is associated with one NBBS trunk.

**Tunnel.** A tunnel is defined by an LNS-LAC pair. The tunnel carries PPP datagrams between the LAC and the LNS. A single tunnel can multiplex many sessions. A control connection operating over the same tunnel controls the establishment, release, and maintenance of all sessions and of the tunnel itself.

**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.

**version.** A separately licensed program that usually has significant new code or new function.

**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 connection.** In frame relay, the return path of a potential connection.

**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.

# Special Characters

**2216 Nways BroadBand Switch.** A fast packet switch enabling high-speed communications over an NBBS Network. The 2220 Nways BroadBand Switch implements the functions defined by the Networking BroadBand Services architecture. Synonymous with ***Nways Switch***.

# Index

## Numerics



## A

## B

# C

# D

# E

# F

# K



# L

# M

# N

## O

# P

# Q

# R



# S

# T



# U

# V



# W



# X

# Readers' Comments — We'd Like to Hear from You

**Nways Multiprotocol Access Services**
**Software User's Guide**
**Version 3 Release 1**

**Publication No. SC30-3886-03**

**Overall, how satisfied are you with the information in this book?**

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|---|---|---|---|---|---|
| Overall satisfaction | ☐ | ☐ | ☐ | ☐ | ☐ |

**How satisfied are you that the information in this book is:**

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|---|---|---|---|---|---|
| Accurate | ☐ | ☐ | ☐ | ☐ | ☐ |
| Complete | ☐ | ☐ | ☐ | ☐ | ☐ |
| Easy to find | ☐ | ☐ | ☐ | ☐ | ☐ |
| Easy to understand | ☐ | ☐ | ☐ | ☐ | ☐ |
| Well organized | ☐ | ☐ | ☐ | ☐ | ☐ |
| Applicable to your tasks | ☐ | ☐ | ☐ | ☐ | ☐ |

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

Company or Organization _____

Phone No. _____

**Readers' Comments — We'd Like to Hear from You**
SC30-3886-03

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

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

IBM Corporation
Design & Information Development
Department CGF/Bldg. 656
PO Box 12195
Research Triangle Park, NC 27709-9990

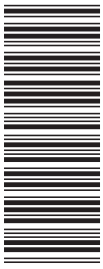
Fold and Tape **Please do not staple** Fold and Tape

SC30-3886-03

Cut or Fold
Along Line

Part Number: 85H7918

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.



85H7918

SC30-3886-03

Spine information:

Nways Multiprotocol Access Services

Nways MAS V3R1 Software User's Guide

SC30-3886-03