

About This Book

Audience • Organization • Related Documentation • Notation

This guide tells you how to manage a network of LightStream 2020 enterprise ATM switches. It is a task-oriented guide that contains procedures for each task. Before describing the network operations tasks, this guide explains how the Simple Network Management Protocol (SNMP) provides the network management information you need to manage your LightStream network. It also tells you how you can manage your LightStream network from a third-party network management system (NMS).

Your network should be fully installed and configured before you attempt to operate it. Refer to the *LightStream 2020 Installation and Troubleshooting Manual* for installation instructions and to the *LightStream 2020 Configuration Guide* for configuration information.

Audience

The *LightStream 2020 Administration Guide* is intended for the manager or advanced user of the LightStream network. This guide provides detailed procedures to help you manage the LightStream network. You should be familiar with the information in the *LightStream 2020 Operations Guide* before attempting to use the procedures in this guide.

The “SNMP Commands” chapter of this guide was written for users who are familiar with standard SNMP commands. If you are not familiar with SNMP, you should not attempt to use the commands described in the “SNMP Commands” chapter.

Users of the LightStream document set are expected to have a general understanding of basic data communications concepts, some knowledge of UNIX, and a familiarity with the interfaces used by the devices connecting to their LightStream network.

It is also recommended that you have a working knowledge of TCP/IP networks. For more information about TCP/IP networks, refer to *Internetworking with TCP/IP, Volume 1, Principals, Protocols, and Architecture* by Douglas E. Comer, 1991, Prentice-Hall, Inc. (ISBN 0-13-468505-9).

Organization

This guide is organized as follows:

- About This Book — Describes the audience, organization, and conventions for this book.
- Before You Begin — Describes the activities that you perform on a LightStream network and the tools that you can use to manage your network.

- **Administrative Tasks** — Explains how to perform basic administrative tasks such as changing the protected mode and npadmin passwords, writing CLI script files, setting the SNMP community, handling CLC, FDDI, and Ethernet interfaces, and handling spanning tree bridging, bridge filters, and VLI internetworking.
- **SNMP Commands** — Describes SNMP procedures and commands and discusses identification of the MIB objects used with the commands.
- **TCS Hub Commands** — Tells you how to access the TCS, the TCS hub user interface and commands.
- **Using LightStream Traps** — Discusses traps generated by LightStream switches, trap types and formats, trap categories and priorities, trap logs, trap settings, and trap displays.
- **Troubleshooting** — Describes trunk line monitoring, isolating trunk problems, diagnosing port problems, and troubleshooting procedures.
- **Optimizing the Load Across Trunks** — Explains how to determine trunk port and connection status and how to manually reroute frame forwarding, frame relay, and ATM UNI connections.

Related Documentation

The following is a list of LightStream manuals and other material relevant to LightStream users.

- *LightStream 2020 System Overview*

The system overview explains what a LightStream switch is and how it works. It outlines ATM technology and describes LightStream hardware and software.
- *LightStream 2020 Site Planning and Cabling Guide*

The site planning and cabling guide (SPCG) tells you how to prepare your site to receive LightStream hardware. It includes space, environmental and electrical requirements, rack selection guidelines, requirements for the management workstation, and information on cables and connectors.
- *LightStream 2020 Installation and Troubleshooting Manual*

The installation and troubleshooting manual (I&TM) tells you how to install LightStream hardware and software, how to diagnose hardware problems, and how to replace faulty hardware components.
- *LightStream 2020 Configuration Guide*

The configuration guide provides the information you need to configure LightStream switches. It describes the configuration tools and how to use them. It describes the configuration database and defines all configurable attributes and their settings. The guide also provides step-by-step configuration procedures.
- *LightStream 2020 Operations Guide*

The operations guide is a task-oriented book that tells you how to operate a network of LightStream switches. The guide presents an overview of network operations tasks, describes the command line interface (CLI), and presents procedures for performing monitor and control tasks such as displaying the status of nodes, cards and ports, viewing statistics, and creating collections of traffic data.
- *LightStream 2020 Traps Reference Manual*

This manual presents an overview of LightStream traps (error and event messages) and a list of operational, SNMP, and informational traps generated by the LightStream switch.

- *LightStream 2020 Command and Attribute Reference Guide*

The reference guide provides detailed descriptions of the syntax and functions of all CLI commands. It also indicates CLI equivalents of configuration procedures, describes the LightStream private MIB, and gives UNIX-style manual pages for selected LynxOS commands.

- *LightStream 2020 Command Line Interface (CLI) Reference Card*

The reference card compactly summarizes the syntax and arguments of all CLI commands.

- *LightStream 2020 Release Notes*

The release notes provide a software upgrade procedure and describe new features and special considerations, including information on known software bugs.

Note The release notes contain important information that does not appear in other documents.

- *LightStream 2020 Online Help*

The LightStream command line interface (CLI) and configuration program both produce online help facilities.

Before attempting to install, configure, operate, or troubleshoot a network of LightStream switches, read the *LightStream 2020 System Overview*. This overview provides important background information about the LightStream product and the ATM technology on which the product is based. After reading the *LightStream 2020 System Overview*, refer to Table 1-1 to determine which manuals you should read next.

Table 1-1 LightStream Reading Path


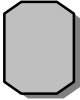





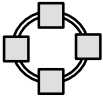

If you want to:	Read the following manuals in the order listed below:
Install LightStream switches	<i>LightStream 2020 Release Notes</i> ¹ <i>LightStream 2020 Site Planning and Cabling Guide</i> <i>LightStream 2020 Installation and Troubleshooting Manual</i>
Configure LightStream switches	<i>LightStream 2020 Release Notes</i> ¹ <i>LightStream 2020 Configuration Guide</i> <i>LightStream 2020 Online Help Screens</i>
Set up or expand a LightStream network	<i>LightStream 2020 Release Notes</i> ¹ <i>LightStream 2020 Administration Guide</i> <i>LightStream 2020 Online Help Screens</i>
Operate a LightStream network	<i>LightStream 2020 Release Notes</i> ¹ <i>LightStream 2020 Operations Guide</i> <i>LightStream 2020 Command and Attribute Reference Guide</i> <i>LightStream 2020 Command Line Interface (CLI) Reference Card</i> <i>LightStream 2020 Traps Reference Manual</i> <i>LightStream 2020 Online Help Screens</i>
Manage or troubleshoot a LightStream network	<i>LightStream 2020 Release Notes</i> ¹ <i>LightStream 2020 Operations Guide</i> <i>LightStream 2020 Administration Guide</i> <i>LightStream 2020 Command and Attribute Reference Guide</i> <i>LightStream 2020 Command Line Interface (CLI) Reference Card</i> <i>LightStream 2020 Traps Reference Manual</i> <i>LightStream 2020 Online Help Screens</i>
Troubleshoot LightStream hardware	<i>LightStream 2020 Release Notes</i> ¹ <i>LightStream 2020 Installation and Troubleshooting Manual</i> <i>LightStream 2020 Site Planning and Cabling Guide</i>

1. We recommend that you review the release notes before attempting to install, configure, operate, or troubleshoot a LightStream switch. The release notes contain important information that does not appear in other documents.

Notation

In this document, several conventions distinguish different types of graphics and text.

Graphics Conventions

This symbol ...	Represents ...
	LightStream ATM Network
	LightStream Switch
	Non-LightStream ATM Device
	Host
	Packet Switch
	Router
	Ethernet LAN
	FDDI LAN
	LAN Bridge

H3300

Text Conventions

Convention	Purpose	Example
Bold screen literal type	Represents user input.	\$ date
Screen literal type	Represents system output	Wed May 6 17:01:03 EDT 1994
Boldface type	Denotes names of commands, command arguments, and switches. Command names are case sensitive; enter them exactly as they appear in the text.	Issue the clear command.

Notation

Convention	Purpose	Example
<i>Italic type</i>	Used for titles of documents and for emphasis.	<i>LightStream 2020 Configuration Guide</i> File names are <i>case</i> sensitive.
Angle brackets < >	Indicate user-specified parameters or classes of user responses. When you see this notation in a syntax statement, make the substitution but do not type the angle brackets.	If you see: set port <c.p> <state> you might type: set port 4.3 active
Square brackets []	Indicate keys on the keyboard, or optional arguments or parameters for commands. You can omit optional arguments and parameters in any command.	Press [Return] . cli> help [<topic>]
Caret symbol ^	When the caret symbol precedes a character, it refers to the control key.	^X is the same as [Control] X
Curly braces { }	Indicate a choice of arguments or parameters for commands. Arguments or parameters are separated by a vertical line {}, and you must select <i>one</i> .	cli> set cli traplevel {off info oper trace debug}

Before You Begin

Where to Begin • Network Management Tasks • Management Tools • Network Management Scenarios

This chapter lists the management activities that you can perform on a network of LightStream® 2020 enterprise ATM switches and describes the tools that you can use to manage your network. It explains the different ways you can manage your LightStream network, depending on your hardware and software, and it also tells you how LightStream switches and the network management tools use the Simple Network Management Protocol (SNMP) to communicate.

Where to Begin

Before you attempt to manage your network, each LightStream switch should be fully installed, powered on, and configured. Refer to the *LightStream 2020 Installation and Troubleshooting Manual* for installation information and to the *LightStream 2020 Configuration Guide* for configuration information. Complete the appropriate setup procedures described in the “Administrative Tasks” chapter of this guide. These procedures allow you to set up the system to run in secure mode and change the default SNMP community and trap delivery addresses. Depending on how you choose to set up your network, you may not need to perform all of the setup procedures.

Network Management Tasks

You can perform a wide variety of network management tasks on your LightStream network. You will perform some every day and others only occasionally. This section lists the different types of network management tasks that you can perform.

Control Tasks

- Configuring a LightStream switch
- Monitoring LightStream switches
- Customizing trap logs and displays
- Logging and viewing traps
- Changing the online network configuration
- Deleting frame relay (FR) data link connection identifiers (DLCIs)
- Deleting ATM-UNI virtual channel identifiers (VCIs)
- Defining and assigning bridge filters
- Handling workgroups
- Creating special files
- Writing command line interface (CLI) scripts
- Performing an orderly shutdown

- Changing the default trap delivery addresses
- Changing the default terminal type

Security-related Tasks

- Setting or changing account passwords
- Setting the SNMP community
- Changing the default modem password and init string
- Creating new user accounts
- Changing the default SNMP community names
- Changing the security level of the LightStream network
- Creating a backup of the network configuration on floppy disks

SNMP Tasks

- Monitoring MIB object values
- Setting and obtaining the values of MIB attributes
- Walking the MIB tree

Troubleshooting Tasks

- Isolating trunk problems
- Looping and unlooping ports and interfaces
- Activating cards, ports, and circuits
- Resetting cards
- Forcing TCS Hub to be primary or secondary
- Loading operational software or diagnostics
- Using the **ping** command
- Verifying the software installation

TCS Hub Tasks

- Connecting to a card
- Forcing a switch card to become the primary or secondary TCS hub

Optimization Tasks

- Determining the status of trunk ports
- Manually optimizing the load across trunks

Management Tools

You can use the following methods to manage a LightStream network:

- Self-management. Use the LightStream configurator and the LightStream monitor to perform most management functions. You can, however, use the CLI on one of the LightStream switches within the network or on a Sun SPARCstation to perform some management tasks, if the configurator is unavailable.
- Third-party management. Use an external, third-party, SNMP-compatible network management system (NMS).

Self-Management Tools

The LightStream Configurator

The LightStream switch comes with the LightStream *configurator*—a user-friendly graphical interface that reduces configuration tasks to the simple click of a mouse button. As network administrator you will use the LightStream configurator (for the most part) to manage a network of LightStream 2020 enterprise ATM switches. See the *LightStream 2020 Configuration Guide* for further details. If the configurator is unavailable to you, you can use the CLI commands in this document to perform many management tasks.

The LightStream Monitor

LightStream technology provides graphical displays of individual LightStream switches, cards, and ports via the LightStream *monitor*. In most instances you will want to monitor the network with the LightStream monitor. See the *LightStream 2020 Operations Guide* for details. However, if the monitor is unavailable to you, you can use the CLI commands in this document to perform many monitoring tasks.

Command Line Interface

The CLI is a simple, line-based interface that runs on a LightStream switch or a Sun SPARCstation. You can access the CLI by connecting a terminal to a LightStream switch, by telnetting to the NP, or by running the CLI on a Sun SPARCstation.

If the LightStream monitor or configurator are unavailable to you and you use the CLI to perform management tasks, any changes you make to configuration attributes may cause the local configuration database to be out of synchronization with the global database.

Third-Party Network Management Tools

You can use any industry standard SNMP-compatible NMS to monitor a LightStream network. The NMS is connected to the LightStream switch via the Ethernet interface on the NP.

The following three systems can be used with the LightStream switch:

- OpenView, Release 3.0 from Hewlett Packard
- SunNet Manager, Release 2.0 from SunConnect
- NetDirector Enterprise Network Management software from Ungermann-Bass

You cannot configure a LightStream network using a third-party NMS. LightStream switches and networks are configured with the LightStream configurator. For information on the LightStream configurator, refer to the *LightStream 2020 Configuration Guide*.

The LightStream documentation set does not provide instructions on how to use a third-party NMS. Refer to the product documentation for your third-party NMS for specific instructions.

Using SNMP-Compatible Tools

As explained in this section, you can manage a LightStream network with its own management software or with an external SNMP-compatible third-party NMS. Regardless of which system you use, the management software communicates with the managed devices using SNMP.

Network Management Scenarios

You can manage your network in a number of different ways, depending on your hardware and software and whether you want traps to be interleaved with, or separated from, your general monitoring and control functions. Traps interleaved with your general monitoring and control functions may interfere with or delay your ability to enter or execute commands or to receive output from the general monitoring and control functions.

Before bringing your LightStream network online, consider how you will monitor and control the network. Table 2-1 lists the different options that are available. Use this table to locate information about the method you wish to use.

The scenarios described here are not mutually exclusive. As long as you have at least one Sun SPARCstation associated with your LightStream network for configuration purposes, you can perform routine monitoring and control tasks on any or all of the systems described in scenarios two through five.

The preferred method of network configuration, monitoring and control is with the LightStream configurator and the Light-Stream monitor running on a Sun SPARCstation as described in scenario 1 in Table 2-1. The user-friendly features of these programs reduce many tasks to the click of a mouse button. LightStream documentation and customer support provide only limited support for scenarios 3, 4, and 5.

Table 2-1 Network Administration Scenarios

No.	Hardware	Software	Interleave Traps?	Reference
1	Sun SPARC-station	Configure, monitor, and control the network on a Sun SPARCstation using LightStream management software. If you cannot access the SPARCstation, you can use CLI on an NP to perform management tasks. (Optionally, other third-party SNMP-compatible network management software can be used.) The configurator, the monitor, and the CLI run and display on the SPARCstation running SunOS 4.1.x. HP OpenView is optional. This scenario is the preferred method for network management tasks.	Yes	“Scenario 1: Manage Network from Sun SPARCstation Using the Configurator, the Monitor, and the CLI”
2	VT100-compatible terminal	After configuring the network using the LightStream configurator on a Sun SPARCstation, monitor and control the network from the VT100 terminal. However, if you must add or move hardware or add ports or VCs, you <i>must</i> use the Sun SPARCstation to run the configurator. The CLI runs on a LightStream network processor (NP) and displays on the VT100.	Yes	“Scenario 2: Manage Network from VT100 Terminal Using the CLI”
3	Sun SPARC-station	After configuring the network using the LightStream configurator, monitor and control the network from the Sun SPARCstation using the CLI and a third-party NMS. The configurator, the CLI, and the third-party NMS trap monitoring tool run and display on the SPARCstation.	No	“Scenario 3: Manage Network from Sun SPARCstation Using the CLI and a Third-Party Trap Monitoring Tool”
4	Non-Sun workstation	After configuring the network using the LightStream configurator on a Sun SPARCstation, monitor and control the network from the non-Sun workstation using the CLI. However, if you must add or move hardware or add ports or VCs, you <i>must</i> use the Sun SPARCstation to run the configurator to complete these tasks. The CLI runs on a LightStream NP and displays on the workstation.	No	“Scenario 4: Manage Network from a Non-Sun Workstation Using the CLI Only”
5	Non-Sun workstation	After configuring the network using the LightStream configurator on a Sun SPARCstation, monitor and control the network from the non-Sun workstation using CLI and a third-party trap monitoring tool. However, if you must add or move hardware or add ports or VCs, you <i>must</i> access the Sun SPARCstation to run the configurator to complete these tasks. The CLI runs on a LightStream NP and displays on the workstation. The third-party NMS trap monitoring tool runs and displays on the workstation.	No	“Scenario 5: Manage Network from a Non-Sun Workstation Using the CLI and a Third-Party Trap Monitoring Tool”

Note You can access traps from a single instance of CLI running on an NP or from one instance of a third-party trap monitoring tool running on a workstation. If you attempt to display traps on a second instance of either program running on another single processor (the NP or workstation), a message is displayed indicating that traps have been intercepted by another user.

Scenario 1: Manage Network from Sun SPARCstation Using the Configurator, the Monitor, and the CLI

Figure 2-1 shows the hardware configuration for scenario 1 and information that might be displayed on the Sun SPARCstation. The two windows in the display show general monitoring and control information and traps.

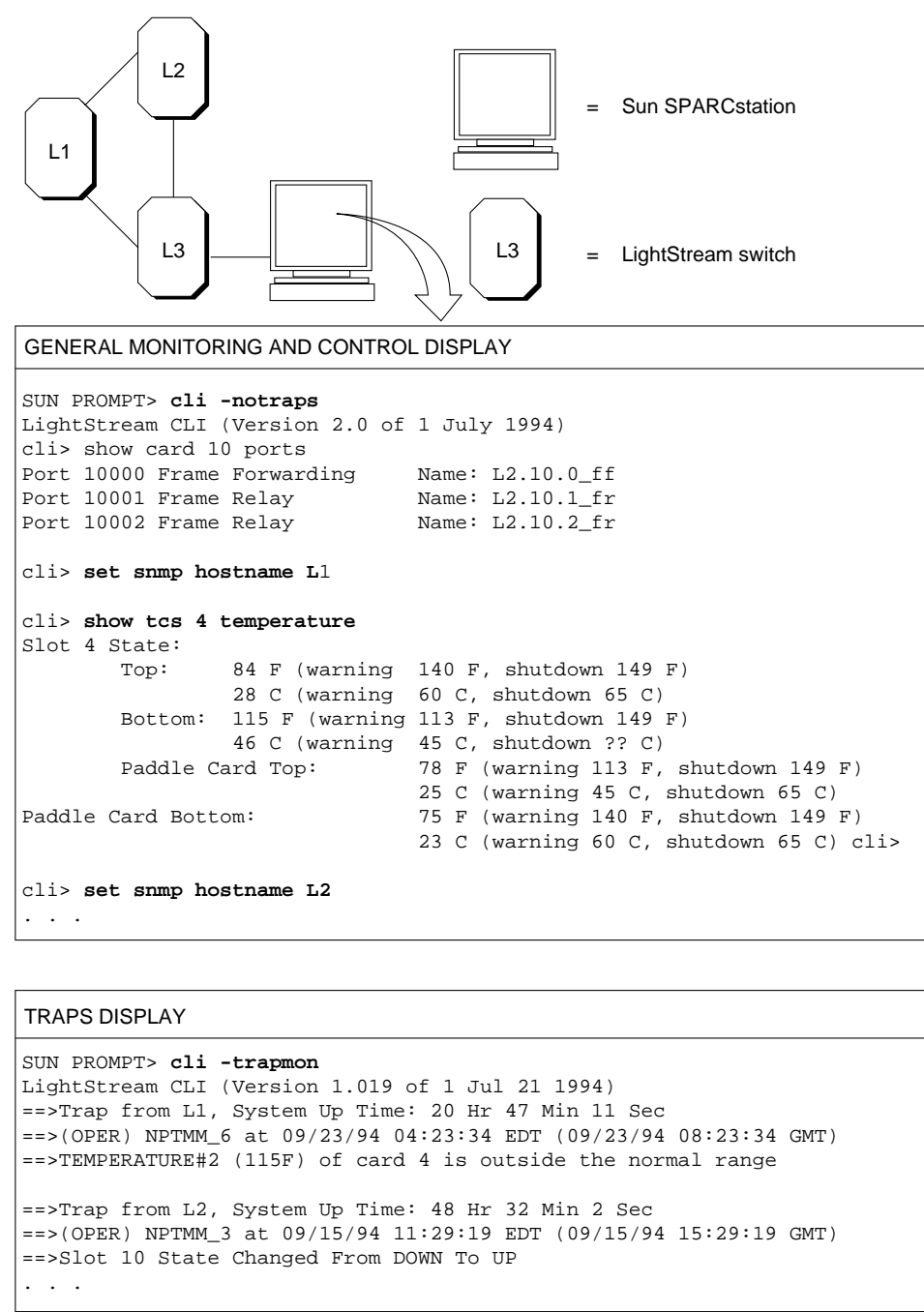
In most instances you will perform management functions with the LightStream configurator and the LightStream monitor running on the SPARCstation. You can invoke the LightStream monitor through HP OpenView to see a graphical representation of LightStream switches, cards, and ports. (The HP OpenView product is not required to run the LightStream monitor.)

You can perform general monitoring and control functions on all LightStream switches in the network by running one instance of the CLI, with traps turned off, in one window of the SPARCstation. You can monitor any other LightStream switch in the network by setting the SNMP hostname to the appropriate switch. (To monitor each LightStream switch in a separate window, run one instance of the CLI with traps turned off for each switch. Each instance of the CLI runs in a separate window on the SPARCstation. This option is not illustrated in Figure 2-1).

You monitor traps for all LightStream switches in the network by running one instance of the CLI with the trap monitoring (trapmon) switch turned on in a window of the SPARCstation. Traps are displayed on the SPARCstation only if you have changed the default trap delivery address in each LightStream switch to the address of the SPARCstation.

General monitoring and control input and output are not interleaved with traps. One window on the workstation is dedicated to traps. One or more windows are dedicated to general monitoring and control.

Figure 2-1 Example of scenario 1



Scenario 2: Manage Network from VT100 Terminal Using the CLI

Figure 2-2 shows the hardware configuration for scenario 2 and information that might be displayed on the VT100 terminal.

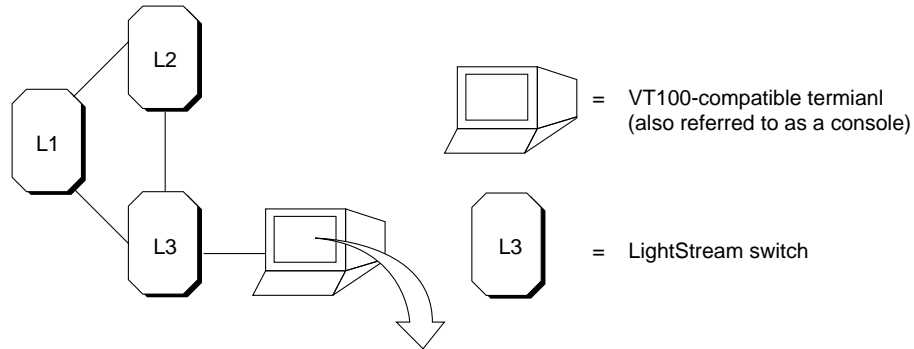
In most instances you will perform many management and monitoring functions with the LightStream configurator and the LightStream monitor running on a SPARCstation. If these tools are unavailable, you can use the CLI procedures described in this manual. However, any changes you make will cause the local database to be out of synchronization with the global database.

You perform general monitoring and control functions on the LightStream switch connected to the console (VT100 terminal) by running the CLI on the NP of that switch. You can monitor and control any other LightStream switch in the network by setting the SNMP hostname to the switch you want to monitor.

When you run the CLI to perform general monitoring and control functions on a particular LightStream switch, trap monitoring for that switch is provided as part of the general monitoring function (unless you have explicitly disabled the display of traps). Two copies of the traps generated on the switch connected to the console are displayed. (One copy is displayed by running the CLI; the other, by an automatic filtering mechanism that displays traps to a local console.) To prevent the duplicate display of traps, use the **set chassis consoletraplevel** command to turn off the filtering mechanism. To monitor traps of other switches in the network, change the default trap delivery addresses for each switch to the IP address of the NP from which the CLI is being run.

Traps are interleaved with general monitoring and control input and output on the VT100 terminal when you use this method.

Figure 2-2 Example of scenario 2



```
cli> show card 10 ports
Port 10000 Frame Forwarding Name: L2.10.0_ff
Port 10001 Frame Relay Name: L2.10.1_fr
Port 10002 Frame Relay Name: L2.10.2_fr

==>Trap from L1, System Up Time: 20 Hr 47 Min 11 Sec
==>(OPER) NPTMM_6 at 09/23/94 04:23:34 EDT (09/23/94 08:23:34 GMT)
==>TEMPERATURE#2 (115F) of card 4 is outside the normal range

cli> set snmp hostname L1

cli> show tcs 4 temperature
Slot 4 State:
  Top:      84 F (warning 140 F, shutdown 149 F)
            28 C (warning 60 C, shutdown 65 C)
  Bottom:   115 F(warning 113 F, shutdown 149 F)
            46 C (warning 45 C, shutdown ?? C)
Paddle Card Top: 78 F (warning 113 F, shutdown 149 F)
                25 C (warning 45 C, shutdown 65 C)
Paddle Card Bottom: 75 F (warning 140 F, shutdown 149 F)
                   23 C (warning 60 C, shutdown 65 C) cli>

==>Trap from L2, System Up Time: 48 Hr 32 Min 2 Sec
==>(OPER) NPTMM_3 at 09/15/94 11:29:19 EDT (09/15/94 15:29:19 GMT)
==>Slot 10 State Changed From DOWN To UP

cli> set snmp hostname L2
. . .
```

S3722

Scenario 3: Manage Network from Sun SPARCstation Using the CLI and a Third-Party Trap Monitoring Tool

Figure 2-3 shows the hardware configuration for scenario 3 and information that might be displayed on the Sun SPARCstation. The two windows in the display show general monitoring and control information and traps.

In most instances you will many perform management and monitoring functions with the LightStream configurator and the Light-Stream monitor running on the SPARCstation. If these tools are unavailable, you can use the CLI procedures described in this manual. However, any changes you make will cause the local database to be out of synchronization with the global database.

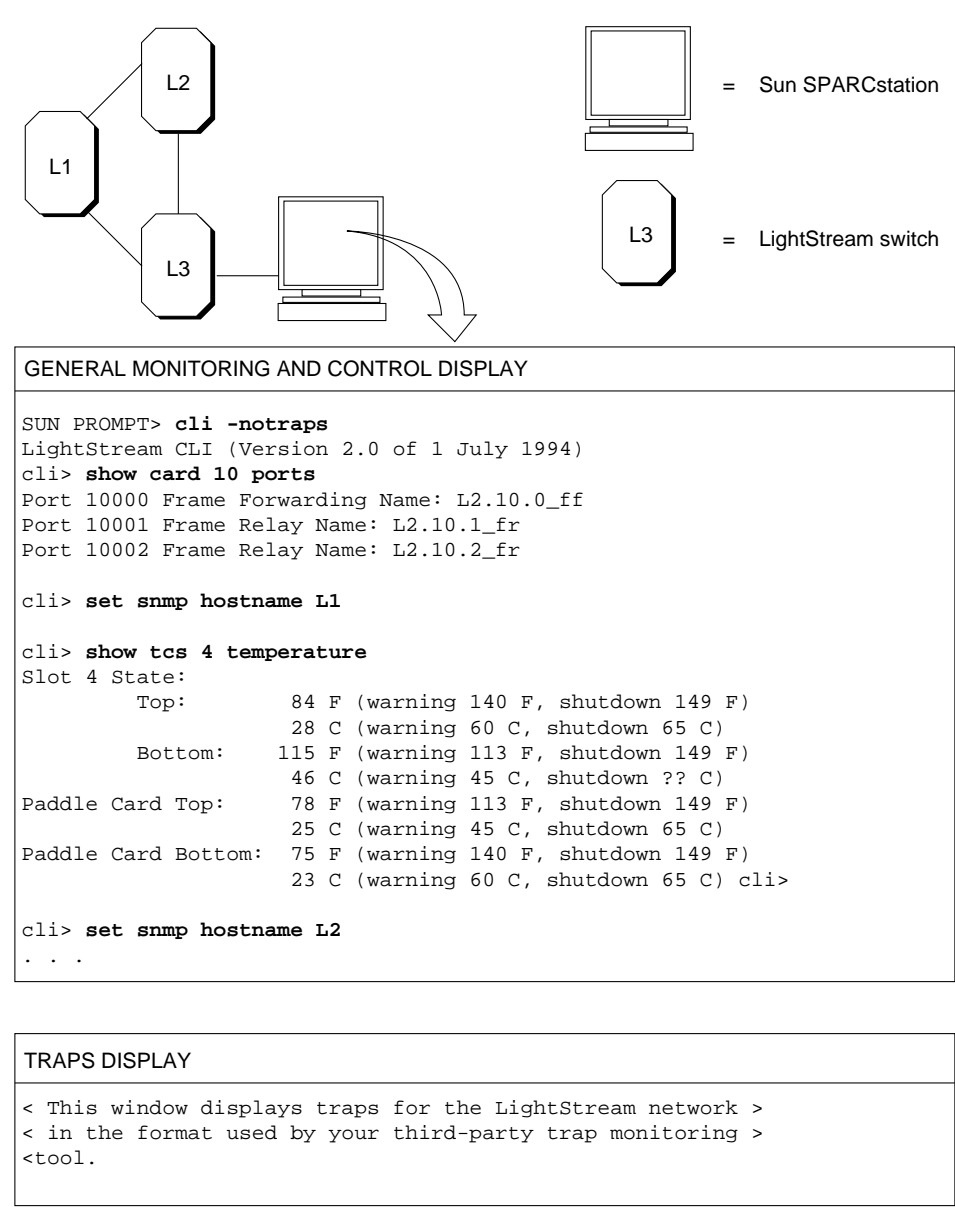
You perform general monitoring and control functions on all LightStream switches in the network by running one instance of the CLI, with traps turned off, in one window of the Sun SPARCstation. You can monitor any other LightStream switch in the network by setting the SNMP hostname to the

appropriate switch. (To monitor each LightStream switch in a separate window, run one instance of the CLI with traps turned off for each switch. Each instance of the CLI runs in a separate window on the SPARCstation. This option is not illustrated in Figure 2-3).

You monitor traps for all LightStream switches in the network by running the trap monitoring tool provided by the NMS to display traps in one window of the SPARCstation. Traps are displayed on the Sun SPARCstation only if you have changed the default trap delivery address in each LightStream switch to the address of the SPARCstation.

General monitoring and control input and output are not interleaved with traps. One window on the workstation is dedicated to traps (using the NMS trap monitoring tool). One or more windows are dedicated to general monitoring and control.

Figure 2-3 Example of scenario 3



Scenario 4: Manage Network from a Non-Sun Workstation Using the CLI Only

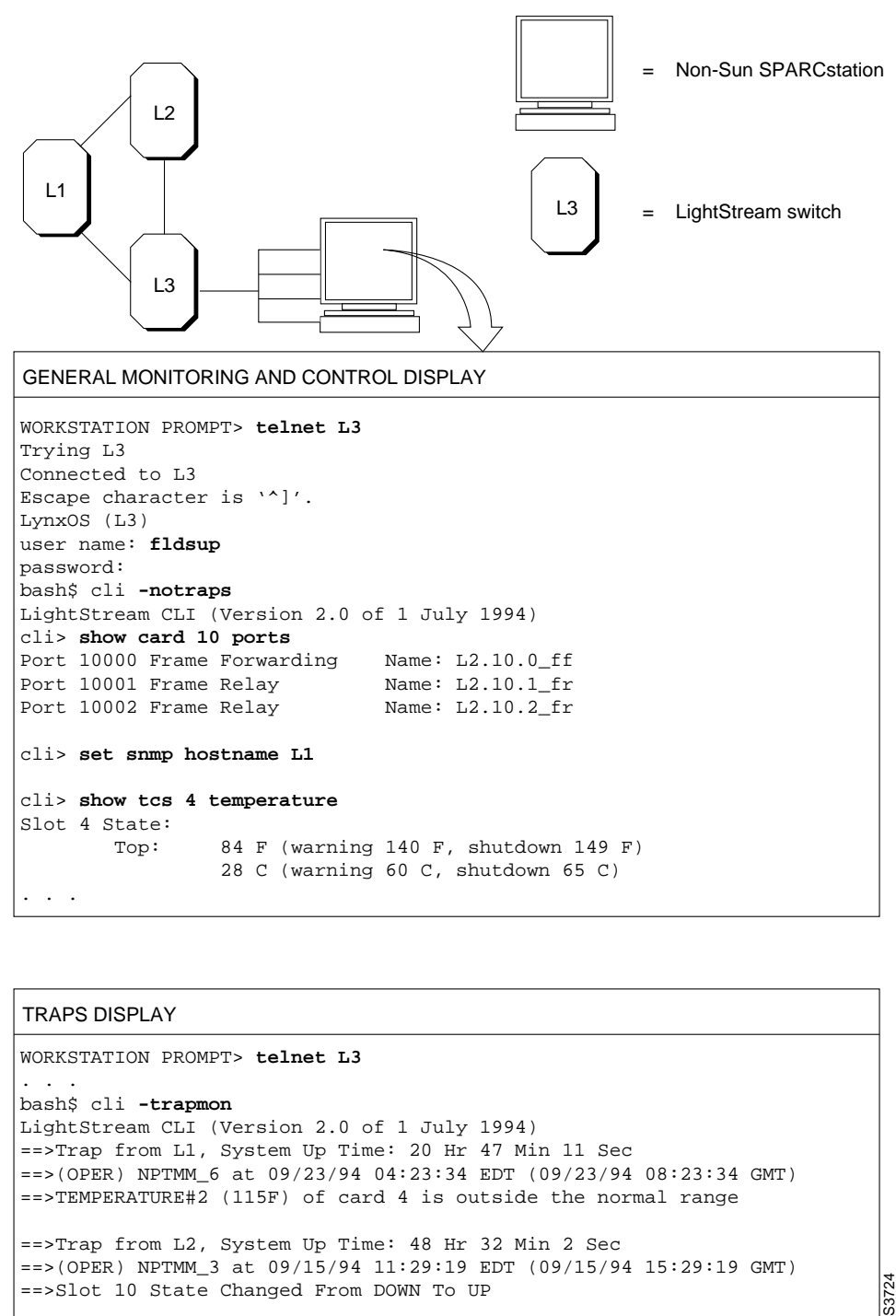
Figure 2-4 shows the hardware configuration for scenario 4 and information that might be displayed on the non-Sun workstation. The two windows in the display show general monitoring and control information and traps.

After configuring your network with the LightStream configurator running on a Sun SPARCstation, you can monitor and control the network from the non-Sun workstation using CLI. You perform general monitoring and control functions on the LightStream switch connected to the non-Sun workstation by telnetting from the workstation to an NP on a LightStream switch and running one instance of the CLI, with traps turned off, on that NP. The CLI display appears in one window on the workstation. You can monitor any other LightStream switch in the network by setting the SNMP hostname to the appropriate switch. (To monitor each LightStream switch in a separate window, run one instance of the CLI, with traps turned off, on the NP of each switch and display it in a separate window on the workstation. This option is not illustrated in Figure 2-4).

You monitor traps for all LightStream switches in the network by telnetting from the workstation to an NP on a LightStream switch and running one instance of the CLI with the trap monitoring (trapmon) switch turned on for that NP. The trap display appears in one window on the workstation. Traps are displayed on the workstation only if you have changed the default trap delivery address in each LightStream switch to the address of the NP on which the trap monitoring tool is running.

General monitoring and control input and output are not interleaved with traps. One window on the workstation is dedicated to traps. One or more windows are dedicated to general monitoring and control.

Figure 2-4 Example of scenario 4



S3724

Scenario 5: Manage Network from a Non-Sun Workstation Using the CLI and a Third-Party Trap Monitoring Tool

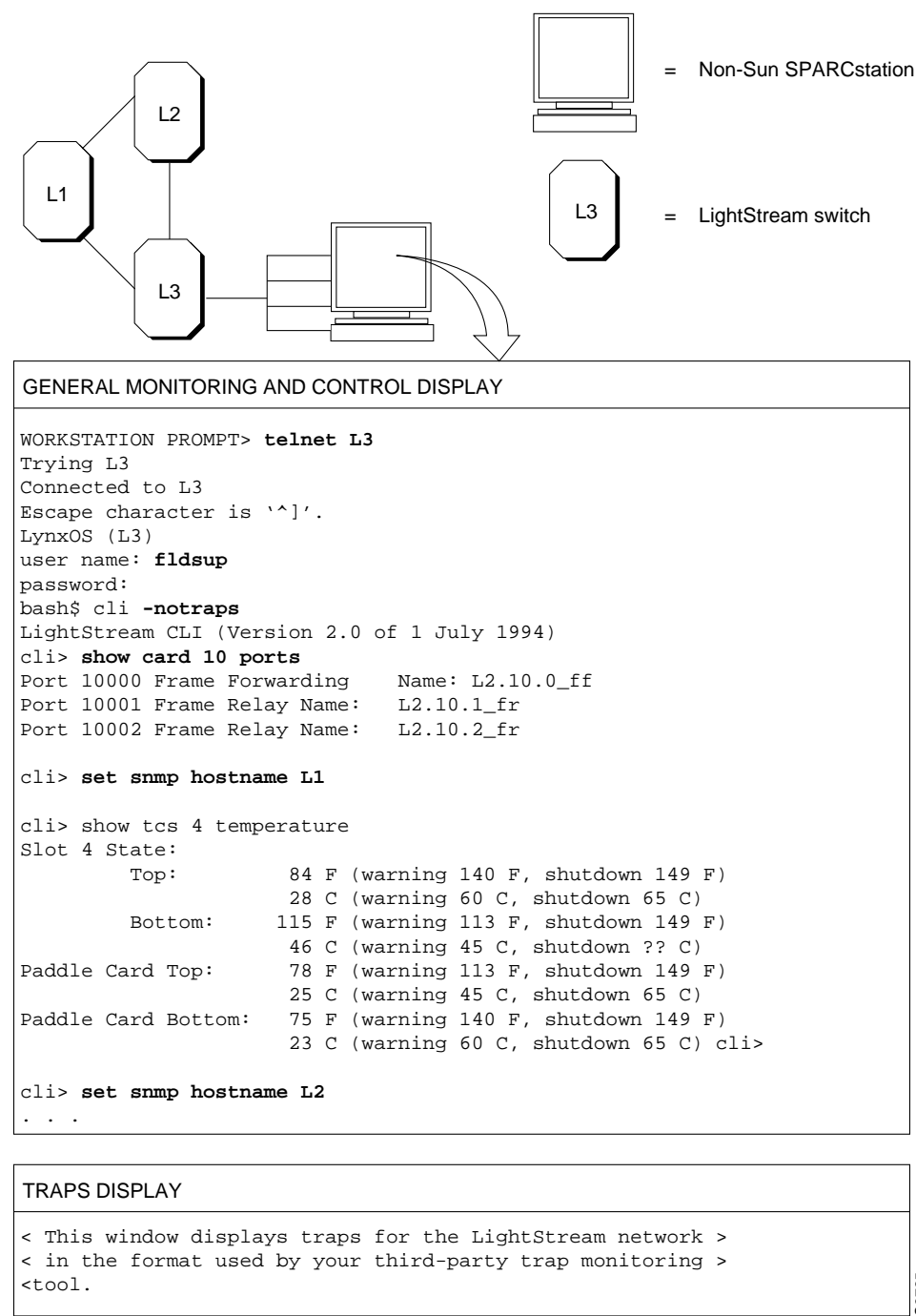
Figure 2-5 shows the hardware configuration for scenario 5 and information that might be displayed on the non-Sun workstation. The two windows in the display show general monitoring and control information and traps.

After configuring your network with the LightStream configurator on a Sun SPARCstation, you can monitor and control the network from the non-Sun workstation using the CLI and a third-party trap monitor. You perform general monitoring and control functions on the LightStream switch connected to the non-Sun workstation by telnetting from the workstation to an NP on a LightStream switch and running one instance of the CLI, with traps turned off, on that NP. The CLI display appears in one window on the workstation. You can monitor any other LightStream switch in the network by setting the SNMP hostname to the appropriate switch. (To monitor each LightStream switch in a separate window, run one instance of the CLI, with traps turned off, on the NP of each switch and display it in a separate window on the workstation. This option is not illustrated in Figure 2-5).

You monitor traps for all LightStream switches in the network by running the trap monitoring tool provided by the NMS to display traps in one window on the workstation. Traps are displayed on the non-Sun workstation only if you have changed the default trap delivery address in each LightStream switch to the address of the NP on which the CLI is running.

General monitoring and control input and output are not interleaved with traps. One window on the workstation is dedicated to traps (using the NMS trap monitoring tool). One or more windows are dedicated to general monitoring and control.

Figure 2-5 Example of scenario 5



Administrative Tasks

Set-up Procedures • Setting Configuration Attributes • Administrative Procedures • Handling Cell Line Card Interfaces • Handling FDDI Interfaces • Handling Ethernet Interfaces • Handling Spanning Tree Bridging • Handling Bridge Filters • Handling Virtual LAN Internetworking

This chapter describes set-up procedures that you should perform before your network becomes active. The set-up procedures include:

- Creating new user accounts
- Creating initialization disks for each LightStream® switch
- Changing the default SNMP community names
- Changing the default trap delivery addresses
- Changing the default terminal type

All of these set-up procedures may not be required for every network. Before you begin, review the procedures to determine which ones apply to your network.

This chapter also describes basic administrative procedures that you may need to perform and details on changing configuration attributes.

Basic Administrative Procedures

The basic administrative procedures include:

- Setting or changing account passwords, protected mode password, and the default modem password and initialization string
- Changing the SNMP community
- Writing /etc/hosts, cli.group, and command line interface (CLI) script files
- Performing an orderly system shutdown
- Backing up configurations
- Handling cell line card (CLC), FDDI, and Ethernet interfaces
- Handling spanning tree bridging
- Working with filters
- Changing the online network configuration
- Deleting a frame relay data link connection identifier (DLCI) or ATM-UNI virtual channel identifier (VCI)

Changing Configuration Attribute Values with CLI Commands

The preferred method of setting attribute values in the local configuration database is by using the LightStream configurator on the network management station (NMS). (See the *LightStream 2020 Configuration Guide* for details.) If the configurator is unavailable, CLI commands may be used to

change configuration attribute values. Normally, configuration changes made with CLI commands are made to run-time memory only. This is useful for testing purposes. However, the next time the node is reset, these volatile changes are overwritten.

CLI commands may also be used to change the local configuration database. CLI set commands write to the local configuration database as well as to run-time memory, if you enter the **set config lock** command first. In this case, your permanent changes are written to the configuration database. If the node is reset, your changes will take effect again.

The **set config lock** command also locks the master management agent (MMA) so that a CLI user connected from a different IP address cannot concurrently make configuration changes. If other users attempt to use CLI set commands while the MMA is locked, they see the error message:

```
SNMP error
```

The CLI issues a periodic reminder that the MMA is locked.

After completing your permanent changes, use the **set config unlock** command to unlock the MMA. This restores the default state (unlocked, no writes to database; other users have access). The lock is automatically removed after 2 minutes of inactivity or when the CLI session is closed.

These commands (**set config lock** and **set config unlock**) are equivalent to setting the `mmaSetLock` MIB object to 3 (locked) or to 1 (unlocked). A third value of the `mmaSetLock` object exists that cannot be set by the **set config** command. If you are testing configuration changes, but do not want to save them yet in the local configuration database, use the **setsnmp mmaSetLock.0 2** command to lock the MMA *without* causing CLI sets to write to the local database.

For additional details about these commands, refer to the *LightStream 2020 Command and Attribute Reference Guide*.

Set-up Procedures

After you complete the installation and startup of a LightStream switch, log in to either the `fldsup` or root user accounts on the switch and perform the set-up procedures described in this section. You need to perform only those set-up procedures that are required for your network.

Some of these procedures require that you restart the (MMA). Each procedure requiring a restart includes that step as part of its process. However, you can perform all of those procedures without restarting the MMA and restart the MMA only once, after you complete the last procedure.

Creating New User Accounts

This section tells you how to create a new user account. The LightStream switch provides an **adduser** script to simplify the task of adding a user to the system.

Procedure

Step 1 Log in to the root account on your LightStream switch.

Step 2 To start the **adduser** script, enter the following at the `bash#` prompt:

```
bash# adduser
```

Step 3 Enter the login name for the new user when you see the following prompt:

```
Enter login name, must be < = 8 characters:
```

Step 4 Enter the full name for the new user when you see the following prompt:

```
Enter user's full name:
```

The program displays login account information as follows:

```
Login Name:<login>
User ID:65536
Home Directory:/usr/<login>
Password Entry: <login>::<UID>:<GID>:<username>:/usr
<login>:/bin/bash
```

where

<login> = The login name of the user.

<UID> = The user identification number.

<GID> = The group identification number.

<username> = The full name of the user.

Step 5 If the information displayed is correct, respond yes (Y) to the following prompt:

```
Add the new user to the password database (Y/N)
```

Step 6 Enter a password for the new user when you see the following prompt:

```
Enter:Adding entry to the /etc/passwd database
Making /usr/<login> home directory
Changing password for <login>
Enter new password:
```

The password must be at least six alphanumeric characters long and must not be a recognized word.

Step 7 Retype the password at the following prompt:

```
Retype new password:
```

The bash# prompt is displayed.

Expected Result

If you enter the new password correctly, the system changes the password and displays the bash# prompt.

A new account with the characteristics you specified is created. You can now log in to the new account and begin using it.

Changing the Default SNMP Community Names

Each LightStream switch has a file detailing each switch in the network that has *read* or *read/write* access to its MMA and the several levels of privileges for each of those switches. To monitor the network, you need only read access to an MMA; however, to make changes to the values in an MMA or to issue control commands, you need read/write privileges.

The software maps the SNMP community name and the IP address of each LightStream switch to a set of privileges. Each LightStream switch has a default file named `/usr/app/base/config/mma.communities` that contains details on the SNMP communities that have been defined for it. A sample of the file is shown in Figure 3-1. The lines preceded by the number sign (#) are informational comments. The last three lines contain the names of the SNMP communities (public, trap, and write).

Figure 3-1 Sample `mma.communities` file

```
bash# more mma.communities
# This is the session configuration file that determines who may
# access the gateway. Each line consists of three items:
# 1st, the session name.
# 2nd, the IP address of the remote site. If address is 0.0.0.0, any
# address may communicate on that session name.
# 3rd, the privileges given that session name. These currently
# consist of READ for read only, WRITE for read/write, or NONE to
# lock out a session name.
# The format is
# session_name IP_address_in_dot_notation privileges
public 0.0.0.0 read
trap 127.0.0.1 write
write 0.0.0.0 write
bash#
```

H3706

The line reading `public 0.0.0.0 read` indicates that a user issuing commands from any IP address (indicated by the notation `0.0.0.0`) and who has set the SNMP community name to *public* has read access to the MMA on this switch. The line reading `trap 127.0.0.1 write` indicates that a user issuing commands from this local switch (indicated by the notation `127.0.0.1`) and who has set the SNMP community name to *trap* has read/write access to the MMA on this switch. The line reading `write 0.0.0.0 write` indicates that a user issuing commands from any IP address (`0.0.0.0`) who has set the SNMP community name to *write* has read/write access to the MMA of this switch.

SNMP community names can be used to provide a level of security to each LightStream switch. It is recommended that you change the names of the *trap* and *write* SNMP communities. By changing the names of the read/write SNMP communities, you can restrict access to only the users who know your SNMP community names. As a convention, most SNMP devices have a community name of *public* with read only privileges. You should not change the name of the public SNMP community, but you can change its privileges, if necessary.

The SNMP communities are defined in the file `/usr/app/base/config/mma.communities`. The following procedure tells you how to edit that file.

Note The SNMP community name is reset to *public* whenever you reset the CLI.

Note The LightStream software contains a number of symbolic links between files in different areas. When a new software release is downloaded, the software looks for *unlinked* files and carries those files forward. Therefore, you must make sure that the files you edit are not linked to any other files in the system. Follow the procedure exactly to ensure that the file you are editing will remain part of the system, even if a new software release is loaded.

Procedure to Change the Default SNMP Community Names

Step 1 Log in to the root account on your LightStream switch.

- Step 2** To gain access to the directory containing the files you want to edit, enter the following at the bash# prompt:

```
bash# cd /usr/app/base/config
```

- Step 3** Move the `mma.communities` file to another file to maintain the link by typing:

```
bash# mv mma.communities mma.communities.orig
```

This means that `mma.communities.orig` is linked to the file `/usr/app/dist/base-x.x.x/config/mma.communities`. (This file is overwritten with the next software upgrade.)

- Step 4** Copy the contents of the file with the link to the original file name, by typing:

```
bash# cp mma.communities.orig mma.communities
```

Now you have two files with identical information. The `mma.communities` file has no links to any other files and the `mma.communities.orig` file is linked to `/usr/app/dist/base-x.x.x/mma.communities`.

- Step 5** Edit the `mma.communities` file using the vi editor, by typing:

```
bash# vi mma.communities
```

If you are not familiar with the vi editor, refer to the *LightStream 2020 Command and Attribute Reference Guide*. When you have edited the file, save the changes.

- Step 6** When you have finished with the file, exit the vi editor by pressing the [Esc] key or ^[followed by typing:

```
ZZ
```

- Step 7** Repeat Step 1 through Step 6 for every LightStream switch in the network.

- Step 8** After you have created the `mma.communities` file for every switch in your network, you should restart the MMA in each switch to update the MMA with the new `mma.communities` info. There are two methods for restarting the MMA.

- If you are restarting the MMA from the bash# prompt, find the process ID (PID) number of the MMA by typing the following at the bash# prompt:

```
bash# ps -ax
```

This lists all processes that are running on the switch. Once you know the PID number for the MMA, enter the following at the bash# prompt:

```
bash# kill -hup <mma pid #>
```

where

<mma pid #> = The PID number for the MMA process in this switch.

This command kills and then restarts the MMA in the target switch.

- If you are restarting the MMA from a CLI running on the switch, find the PID of the MMA by typing the following at the cli> prompt:

```
cli> walk pidName
```

Once you know the PID for the MMA, enter the following at the cli> prompt:

```
cli> set pid <mma pid #> adminstatus inactive
```

This command stops the MMA. The NDD process automatically restarts the MMA.

Expected Results

You can look at the edited `mma.communities` file by entering **more mma.communities** at the `cli>` prompt. The procedure outlined in the section “Changing the Default SNMP Community Names” describes how to change from one SNMP community to another.

Changing the Trap Delivery Address

When you start the CLI, the LightStream switch finds the addresses for trap delivery in the `/usr/app/base/config/mma.trap_communities` file. By default all LightStream switches send traps to their local network processor (NP). However, to have a single CLI or a third-party network management system (NMS) collecting all traps for the network, you must add its address to this file. This change should be made at startup time, because the LightStream switch must be restarted after the file is changed.

The procedure below tells you how to edit the file.

Each line in the `/usr/app/base/config/mma.trap_communities` file consists of the following three items:

- Community name
- IP address for delivery of the traps
- User Datagram Protocol (UDP) port on which traps are sent

SNMP normally uses UDP port 162 for traps. The following is a sample `/usr/app/base/config/mma.trap_communities` default file:

```
trap 127.0.0.1 162
```

Note The LightStream software contains a number of symbolic links between files in different areas. When a new software release is loaded, the software looks for *unlinked* files and carries those files forward. Therefore, you must make sure that the files that you edit are not linked to any other files in the system. Follow the procedure below *exactly* to ensure that the file you are editing will remain part of the system, even if a new release of the software is loaded.

Procedure to Change the Trap Delivery Addresses

Step 1 Determine where you want the traps to be sent for each LightStream switch before you edit the `/usr/app/base/config/mma.trap_communities` file.

Step 2 Log in to the root account on the LightStream switch.

Step 3 To gain access to the directory containing the files you want to edit, enter the following at the `bash#` prompt:

```
bash# cd /usr/app/base/config
```

Step 4 Move the `mma.trap_communities` file to another file to maintain the link by typing:

```
bash# mv mma.trap_communities mma.trap_communities.orig
```

This means that `mma.trap_communities.orig` is now linked to the `/usr/app/dist/base-x.x.x/config/mma.trap_communities` file. (This file is overwritten with the next software upgrade.)

Step 5 Copy the contents of the linked file by typing:

```
bash# cp mma.trap_communities.orig mma.trap_communities
```

Now you have two files with identical information. The `mma.trap_communities` file has no links to any other files and the `mma.trap_communities.orig` file is linked to `/usr/app/dist/base-x.x.x/mma.trap_communities`.

Step 6 Invoke the vi editor to edit the file, by typing:

```
bash# vi mma.trap_communities
```

Refer to the *LightStream 2020 Command and Attribute Reference Guide* for more information on the vi editor. When you have completed the changes, save the file.

Step 7 When you have finished entering the group names and trap numbers in the file, exit the vi editor by pressing the **[Esc]** key or **^[** followed by typing:

```
ZZ
```

Step 8 Repeat Step 2 through Step 7 for every LightStream switch in the network.

Step 9 After you have created the `mma.trap_communities` file for every switch in your network, restart the MMA in each switch by one of the following methods:

- If you are restarting the MMA from the `bash#` prompt, find the PID number of the MMA by typing

```
bash# ps -ax
```

This lists all processes that are running on the switch. Once you know the PID number for the MMA, enter the following at the `bash#` prompt:

```
bash# kill -hup <mma pid #>
```

where

<mma pid #> = The pid number for the MMA process in this switch.

This command kills the MMA in the target switch. The NDD process automatically restarts the MMA.

- If you are restarting the MMA with CLI, find the PID number of the MMA by typing the following at the `cli>` prompt:

```
cli> walk pidName
```

Once you know the PID number for the MMA, enter the following at the `cli>` prompt:

```
cli> set pid <mma pid #> adminstatus inactive
```

This command stops the MMA and the NDD process automatically restarts the MMA.

Expected Results

You can look at the edited `mma.trap_communities` file by typing **more mma.trap_communities** at the `cli>` prompt.

Changing the Default Terminal Type

Whenever you log in to the CLI, the default terminal type of each user account (oper, npadmin, fldsup, and root) is set to vt100. If you do not use a VT100 terminal, you may want to change the default terminal type in your .profile file, so that you need not change the terminal setting each time you log in. This procedure changes the default terminal type setting in the .profile file for each user account.

Procedure 1: Changing the Terminal Type from the bash Prompt

- Step 1** Verify that the terminal type you want to use is defined in the /etc/termcap file.
- Step 2** Log in to the fldsup account or the root account for the LightStream switch.
- Step 3** To edit the terminal type for the oper account, enter the following at the bash# prompt:

```
bash# vi /usr/oper/.profile
```

The vi editor opens and you are ready to edit the default terminal type.

- Step 4** Change the default terminal type for the oper account by editing the line that reads:

```
TERM=vt100
```

to reflect the terminal type you are using. (The terminal type must be defined in the /etc/termcap file.)

If the line TERM=vt100 does not appear in the .profile file, add a line in the following format to that file:

```
TERM=<your default terminal type>
```

- Step 5** Exit from the vi editor by typing the [Esc] key or ^[followed by typing:

```
ZZ
```

- Step 6** Repeat Step 3 through Step 5 for each of the default login accounts by editing the following files:

```
/usr/npadmin/.profile  
/usr/fldsup/.profile  
/usr/root/.profile
```

- Step 7** Repeat Step 3 through Step 5 for any other accounts that you have created in addition to the default user accounts.

Expected Result

The new terminal type does not take effect until you restart the CLI. To make these changes effective immediately, exit from the CLI, then restart the CLI.

Procedure 2: Changing the Terminal Type from the CLI

- Step 1** Verify that the terminal type you want to use is defined in the /etc/termcap file.

- Step 2** At the cli> prompt, type:

```
cli> protected
```

- Step 3** Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 4 To edit the terminal type for the oper account, enter the following at the *cli> prompt:

```
*cli> shell "vi /usr/oper/.profile"
```

The vi editor opens and you are ready to edit the default terminal type.

Step 5 Change the default terminal type for the oper account by editing the line that reads:

```
TERM=vt100
```

to reflect the terminal type you are using. (The terminal type must be defined in the /etc/termcap file.)

If the line TERM=vt100 does not appear in the .profile file, add a line in the following format to that file:

```
TERM=<your default terminal type>
```

Step 6 Exit from the vi editor by typing the [ESC] key or ^[followed by typing:

```
ZZ
```

Step 7 Repeat Step 4 through Step 6 for each of the default login accounts by editing the following files:

```
/usr/npadmin/.profile  
/usr/fldsup/.profile  
/usr/root/.profile
```

Step 8 Also repeat Step 4 through Step 6 for any user accounts you have created in addition to the default accounts.

Expected Result

The new terminal type does not take effect until you restart the CLI. To make these changes effective immediately, exit from the CLI, then restart the CLI.

Editing the hosts File

As the network administrator, you must maintain the `usr/etc/hosts` file on each LightStream switch in your network. This file stores the names and addresses of all LightStream switches in your network. It is created at installation, but you must add an entry for each LightStream switch in your network.

Note Do not modify the existing contents of the `/usr/etc/hosts` file. These entries refer to the local switches. You should only add entries for other devices in your network.

If you are changing the *target* switch for a CLI command and you want to verify the IP address of that switch, you can display the `/etc/hosts` file to check the address.

Follow the procedure below to edit the `usr/etc/hosts` file.

Procedure to Edit the hosts File

Step 1 Log in to the LightStream switch and start the CLI.

Step 2 Type **protected** to invoke protected mode and enter the protected mode password at the prompt.

Step 3 Enter the following:

```
*cli> shell bash
```

You will see a bash prompt.

Step 4 Move to the /usr/etc directory by entering the following at the bash# prompt:

```
bash# cd /usr/etc
```

Step 5 At the bash# prompt, enter:

```
bash# vi hosts
```

The vi editor opens on your screen. If you are not familiar with the vi editor, refer to the *LightStream 2020 Command and Attribute Reference Guide*.

Step 6 Go to the end of the file and enter the names of the LightStream switches in your network and their IP addresses in the file, using the format shown in the “Expected Results” section below.

Step 7 When you have finished entering the names and addresses in the file, exit the vi editor by pressing the [ESC] key or ^[followed by typing:

```
ZZ
```

Step 8 To return to the CLI, enter the following at the bash# prompt:

```
bash# exit
```

Step 9 To exit protected mode, enter the following at the *cli> prompt:

```
*cli> exit
```

Expected Results

A usr/etc/hosts file might look like this:

```
127.1.22.41 Light1
127.1.22.42 Light2
127.1.22.43 Light3
127.1.22.46 Light4
127.0.0.1 localhost
```



Caution The /usr/etc/hosts file on each switch contains chassis-specific information that is entered automatically and modified each time the system boots. Do not copy this file from one LightStream switch to another because the chassis-specific information will be wrong.

Setting Configuration Attributes

This section lists the attributes that you can configure with the LightStream configuration tool on a workstation, and shows the functionally equivalent CLI commands for configuring the attributes, if the configurator is not available to you. (See the *LightStream 2020 Command and Attribute Reference Guide* for more information on these attributes and commands.)

Configuration Attribute Storage Locations

Configuration information is stored in five locations:

- Step 1** Pre-programmed defaults.
- Step 2** Run-time memory on the target node.
- Step 3** A copy in on-board EEPROM of the run-time values of certain line card attributes.
- Step 4** The local configuration database on the disk of the target node.
- Step 5** The global configuration database on the network management station (NMS) on which the configuration tool is running.

Changing Attribute Values

The preferred method of setting attribute values in the local configuration database is by using the LightStream configurator. See the *LightStream 2020 Configuration Guide* for details.

Synchronizing Databases

After you make changes to the local configuration database with CLI commands, the local configuration database is out of synchronization with the global database. The global configuration database is located on the NMS on which the configuration tool is running. To copy local configuration changes to the global database, you must use the verify function in the configuration tool. This function retrieves the local settings and allows you to write them over the values in the global database. See the “Getting Started” chapter of the *LightStream 2020 Configuration Guide*.

Working with Temporary Configuration Changes

CLI changes in run-time memory may cause confusion. When the NP is restarted (for example, when the node is rebooted), or, for card or port attributes, when a card comes up, attribute values in run-time memory are reset in the following sequence:

- Pre-programmed defaults are read in to run-time memory.
- Run-time values of some port attributes are read in from on-board EEPROM and overwrite the default settings.
- Values stored in the local configuration database overwrite any prior values.

Note The result can be a combination of database settings, settings stored in EEPROM (which might have been mistakenly thought to be temporary changes in run-time memory), and defaults.

Port Attributes Stored in EEPROM

Run-time values of the following port attributes are stored in EEPROM on the line card:

Line Card Attribute	Card Type
edge/trunk mode	LSC, MSC, CLC
cell payload scrambling	MSC, CLC
clocking (internal/external)	CLC
bitrate	LSC

Line Card Attribute	Card Type
DCE/DTE mode	LSC
cable length	MSC
C-bit Parity/Clear Channel	MSC
HEC/PLCP mode	MSC

Normally, when you set the status of a card to inactive and then to active, changes made with CLI commands to run-time memory (but not to the database) are lost. However, changes to the above attributes are read back into run-time memory from EEPROM when the card comes up. A value read in from EEPROM remains in effect if the default value for that attribute was never changed in the local configuration database.

For example, if you use the CLI **set port *c.p* characteristics** command to specify the DCE bit rate for a port, this attribute is not configured in the local database because you have been using the default bit rate for this port, or perhaps you are configuring a new card with CLI commands. Because this was an experimental setting, you did not first use the **set config lock** command to write this change to the local database. The new run-time value of this attribute is stored in EEPROM on the line card. When the card is disabled and enabled (or the NP restarted), NP software reads EEPROM and writes the bit rate to run-time memory. It remains because nothing exists in the local database to overwrite it.

Node Attributes that Look Like Port Attributes

A similar confusion is due to thinking of certain node attributes mistakenly as edge card and port attributes. These are the attributes concerned with the following LAN-related functions:

- Workgroups
- Filters
- Spanning tree bridge
- Static bridge routes
- The default bridging action for a port

These attribute values are stored in the NP's run-time memory and not on the card. Consequently, they are overwritten from the local configuration database only when the NP is restarted, typically when the node is rebooted. They are not overwritten when a card is disabled and re-enabled, nor even when the card is physically removed and reinstalled.

If you have used the CLI to specify custom filters and assign them to a port without using the **set config lock** command first to write them to the local database, the filter settings are retained in NP memory. When the edge card is disabled and re-enabled, the filter settings are not affected. You should use CLI commands or the configurator to delete them. They are also removed when you reboot the node or in some other way restart the NP.

Configurable Attributes

Table 3-1 contains the attributes that can be changed with the CLI **set** and **setsnmp** commands.

Table 3-1 CLI-configurable Attributes

Attribute Type	Attribute Name	CLI Command
System	Chassis ID	setsnmp chassisId.0 <i>ID#</i>
	Chassis Name	set chassis name <i>name</i>
	Chassis Type	setsnmp sysDescr.0 " <i>Node description</i> "
	Contact	setsnmp sysContact.0 " <i>Contact information</i> "
	Location	setsnmp sysLocation.0 " <i>Location information</i> "
IP Address	Active Management IP Address	set chassis activeip <i>IPaddress</i>
	IP Address Default Router	set chassis defrouter <i>IPaddress</i>
	NP Ethernet IP Address	set chassis ethernetaddr <i>IPaddress</i>
	NP Ethernet IP Mask	set chassis ethernetmask <i>mask</i>
	Secondary Management IP Address	set chassis secondaryip <i>IPaddress</i>
	Switch Port Subnet Mask	set chassis netmask <i>mask</i>
	Trap Level for Console Display	set chassis consoletraplevel { oper info trace debug }
SNMP Agent	Trap Filter	set cli traplevel { oper info trace debug }
	Trap Log Status	set chassis traplog { on off }
Congestion Avoidance (CA)	Maximum Interval for Permit Level Reports	set chassis congestion maxpermitinterval <i>ms</i>
	Minimum Interval for CA Info Reports	set chassis congestion minpermitinterval <i>ms</i>
	Minimum Interval for Permit Level Reports	set chassis congestion mincainfointerval <i>ms</i>
Master Management Agent (MMA)	MMA Data Collection Space	set chassis agent
	MMA Trap Filter	set cli traplevel { oper info trace debug }
	MMA Trap Log Status	set chassis traplog { on off }
Card	Max VCs for This Card	setsnmp cardMaxVCs . <i>card#</i> <i>nnn</i>
	Administrative Status	set card { active inactive testing }
	Card Name	setsnmp cardName . <i>card#</i> <i>name</i>
	Card Type	setsnmp cardBoardType . <i>card#</i> <i>type</i>
(Common) Port	Port Name	setsnmp portInfoName . <i>portID</i> <i>name</i>
	Port Status	set port <i>c.p</i> { active inactive testing }
LSC Port	DCE Bit Rate	set port <i>c.p</i> characteristics dce-bitrate <i>Kbytes</i>
	DCE/DTE Type	set port <i>c.p</i> characteristics dce-dte-type { dce dte dce-internal }
	DTE Bit Rate	set port <i>c.p</i> characteristics dte-bitrate <i>Kbytes</i>
	Err. Threshold	setsnmp frProvMiNetErrorThreshold . <i>portID</i> <i>n</i>
	Full Enquiry Interval	setsnmp frProvMiUserFullEnquiryInterval . <i>portID</i> <i>n</i>
	LMI Type	set port <i>c.p</i> lmiconfig { none frif ansi_t1_617d q933a }

Setting Configuration Attributes

Attribute Type	Attribute Name	CLI Command
	Max Frame Size	setsnmp edgeMaxFrameSize. <i>portID</i> bytes
	MaxSuppVCs	setsnmp frProvMiMaxSupportedVCs. <i>portID</i> n
	MonEvents	setsnmp frProvMiNetMonitoredEvents. <i>portID</i> n
	Net Interface Type	set port <i>c,p</i> netinterfacetype {uni nni}
	Net Req. Interval	setsnmp frProvMiNetRequestInterval. <i>portID</i> sec
	Polling Interval	setsnmp frProvMiUserPollingInterval. <i>portID</i> sec
	User Err. Threshold	setsnmp frProvMiUserErrorThreshold. <i>portID</i> n
T3 and E3 MSC Port	Cable Length	setsnmp ms1InfoAdminCableLength. <i>portID</i> n
	DS3 Line Type (T3 only)	setsnmp dsx3LineType. <i>portID</i> {4 5 6 8} (default = 5 if T3, 6 if E3/G.804, 8 if E3/Scrambling)
	Cell Payload Scrambling	setsnmp ms1InfoAdminScramble. <i>portID</i> {1 2} setsnmp clc1InfoAdminScramble. <i>portID</i> {1 2}
FDDI Port	Link Error Rate Alarm	set port fddi port{master slave}
	Link Error Rate Cutoff	
	Notify Timer	setsnmp fddimibSMTTNotify.fddimibSMTIndex
OC3 Port	Clocking Type	setsnmp clc1InfoAdminClock. <i>portID</i>
	Cell Payload Scrambling	setsnmp clc1InfoAdminScramble. <i>portID</i>
Per-port Call Set-up and Bandwidth	Call Setup Backoff Adjustment	setsnmp edgeCallSetupBackoff. <i>portID</i> n (default = 5)
	Call Setup Retry Period	setsnmp edgeCallSetupRetry. <i>portID</i> sec (default = 5)
Spanning Tree	Enable Spanning Tree Bridge	set port <i>c,p</i> stb enable
	Forward Delay	set stb forwdelay t
	Hello Time	set stb hellotimer t
	Max Age	set stb maxage age
	Priority	set stb priority pri
	Path Cost	set port <i>c,p</i> stb pathcost n
	Port Priority	set port <i>c,p</i> stb priority n
	STB Enabled	set port <i>c,p</i> stb {enable disable}
Custom Filter	B'cast Limit	set port <i>c,p</i> bcast-limit fps
	Card	(The <i>c</i> in syntax of set port <i>c,p</i> commands)
	Default Action	set port <i>c,p</i> bflt-def ID {block forward}
	Filter ID	(The <i>ID</i> in define bflt ID filter-exp)
	Filter Test	define bflt ID filter-expr
	Forward or Block	set port <i>c,p</i> bflt ID {block n forward n}
	ID	set port <i>c,p</i> bflt ID {block n forward n delete}
	Port	(The <i>p</i> in syntax of set port <i>c,p</i> commands)
	Priority	(The <i>n</i> in set port <i>c,p</i> bflt action n)
Static Route	MAC	set stb static MACaddr rcv { <i>c,p</i> any} xmit <i>c,p</i>

Setting Configuration Attributes

Attribute Type	Attribute Name	CLI Command
	Rcv Port	(The <i>rcv</i> argument in a set stb static command)
	Xmit Ports	(The xmit argument in a set stb static command)
SNMP Options	Trap Address	set snmp hostname { <i>name</i> <i>IPaddress</i> }
	Trap Community Name	set snmp community <i>name</i>
PVC Source (Node A) ¹	A DLCI Number	(The <i>dlci</i> # in set port c.p dlci dlci# commands)
	A Insured Rate	set port c.p dlci dlci# insured-rate <i>bps</i> set port c.p frameforwarding insured-rate <i>bps</i> set port c.p atm-vci vci# insured-rate <i>cps</i>
	A Maximum Rate	set port c.p dlci dlci# max-rate <i>bps</i> set port c.p frameforwarding max-rate <i>bps</i> set port c.p atm-vci vci# max-rate <i>cps</i>
	A Node	(Same as B Node, but while connected to other host)
	A Port	(Same as B Port, but while connected to other host)
	A VCI	(Same as B VCI, but while connected to other host)
PVC Destination (Node B) ¹	B DLCI Number	set port c.p dlci A-dlci# destdlci B-dlci#
	B Insured Rate	set port c.p dlci dlci# insured-rate <i>bps</i> set port c.p frameforwarding insured-rate <i>bps</i> set port c.p atm-vci vci# insured-rate <i>cps</i>
	B Maximum Rate	set port c.p dlci dlci# max-rate <i>bps</i> set port c.p frameforwarding max-rate <i>bps</i> set port c.p atm-vci vci# max-rate <i>cps</i>
	B Node	set port c.p dlci dlci# destnode { <i>S/N</i> <i>IPaddr</i> } set port c.p frameforwarding destnode { <i>S/N</i> <i>IPaddr</i> } set port c.p atm-vci vci# destnode { <i>S/N</i> <i>IPaddr</i> }
	B Port	set port c.p dlci dlci# destport <i>c.p</i> set port c.p frameforwarding destport <i>c.p</i> set port c.p atm-vci vci# destport <i>c.p</i>
	B VCI	set port c.p atm-vci vci# destvci <i>destvci#</i>
Per-port Call Set-up and Bandwidth	Destination Insured Burst Size	Same as Source Insured Burst Size, from destination host
	Destination Maximum Burst Size	Same as Source Maximum Burst Size, from destination host
	Source Insured Burst Size	set port c.p dlci dlci# insured-burst <i>bps</i> set port c.p frameforwarding insured-burst <i>bps</i> set port c.p atm-vci vci# insured-burst <i>cps</i>
	Source Maximum Burst Size	set port c.p dlci dlci# max-burst <i>bps</i> set port c.p frameforwarding max-burst <i>bps</i> set port c.p atm-vci vci# max-burst <i>cps</i>
	Principal Type of Service	set port c.p atm-vci bwtype

Attribute Type	Attribute Name	CLI Command
	Transmit Priority	set port <i>c.p</i> atm-vci priority
	User Data Per Cell	setsnmp edgeUserDataPerCell, <i>portID</i> bits
	Secondary Scale	setsnmp edgeSecondaryScale, <i>portID</i> n

1. For PVC attributes, the CLI commands take the source node, port, and VCI number from the target host. You must set the PVC attributes from both ends and then activate the PVC using the **set port active** command.

Administrative Procedures

This section describes how to use the CLI to change configuration attributes. Normally, these changes are made to run-time memory only. (If the node is reset, the changes are overwritten by the attribute settings in the configuration database.)

Changing the Network Configuration in Run-time Memory

You can make run-time changes to test network performance using different attribute settings. Once you find the appropriate settings, you can use the configurator to make permanent changes. You can also use temporary run-time configuration changes to solve network problems.

Procedure 1: Making Configuration Changes with the CLI.

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

A screen similar to the following is displayed:

```
*cli> show snmp

Community: public
HostName: localhost
Authentication: off
cli>
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 Change the attributes as desired, using the information provided in Table 3-1.

Step 3 If you change any port attributes, that port is briefly disabled. Enter the following at the cli> prompt for any reconfigured port:

```
cli> set port <port#> characteristics executechange
```

This command brings the port down, makes the changes, and brings the port back up. (The CLI does not issue a response to the **set** commands.)

where

<port#> = The port number in card.port format (card = 2 - 10; port = 0 - 7).

Expected Result

The values are changed online.

Creating a DLCI

This section tells you how to create a frame relay DLCI. Normally, this procedure changes the runtime environment but does not alter the DLCI information stored on hard disk.

Note To make these changes permanent, issue the command **set config lock** first.

Procedure to Create a DLCI

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To establish the connection to a node, enter the following at the cli> prompt:

```
cli> set port <port#> dlci <dlci#> <destnode> activate
```

where

- <port#> = The frame relay port in card.port format (card = 2 - 10; port = 0 - 7).
- <dlci#> = The DLCI number associated with the connection. The DLCI number must be in the range 16-991, inclusive.
- <destnode> = The destination for the DLCI. Enter either the IP address or the serial number chassis ID of the destination switch.

For example, to establish DLCI number 100, from port 3.0 on your target node, to the node with IP address 198.113.179.13, you would enter the following command:

```
cli> set port 3.0 dlci 100 198.113.179.13 activate
```

If you know the chassis ID (5146) of the target node, you can use the chassis ID instead of the IP address and enter the following command:

```
cli> set port 3.0 dlci 100 5146 activate
```

Step 3 To establish the connection between two ports on the same node, enter the following at the cli> prompt:

```
cli> set port <port#> dlci <dlci#> <destport> activate
```

where

<destport> = The destination port for the DLCI, entered in in card.port format (card = 2 - 10; port = 0 - 7).

In this case, to establish DLCI number 100, from port 3.0 on the target node, to port 4.7 on the same node, you would enter the following command:

```
cli> set port 3.0 dlci 100 4.7 activate
```

Note The CLI does not display a message indicating that the DLCI was created.

Step 4 To verify that the DLCI exists, enter the following at the cli> prompt:

```
cli> show port <port#> listdlci
```

This displays a list of all DLCIs configured on the specified port. The new DLCI should appear in the list.

Deleting a DLCI

This section tells you how to delete a particular DLCI. Deleting a DLCI brings down the connection for which the specified port is an endpoint. Normally, this procedure changes the runtime environment but does not delete the DLCI from the configuration information stored permanently on hard disk.

Note To make this change permanent, issue the command **set config lock** first.

Procedure to Delete a DLCI

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the list of DLCIs currently configured on the port, enter the following at the cli> prompt:

```
cli> show port <port#> listdlci
```

Step 3 At the cli> prompt, enter the following:

```
cli> set port <port#> dlci <dlci #> delete
```

where

- <port#> = The port number in card.port format (card = 2 - 10; port = 0 - 7) of the port at one end of the DLCI. (You only need to perform the delete action at one end of the DLCI to eliminate it.)
- <dlci #> = The number of the DLCI you want to delete. The number is in the range 16 - 991.

Note The CLI does not display a message indicating that the DLCI was deleted.

Step 4 To verify that the DLCI was deleted, enter the following at the cli> prompt:

```
cli> show port <port#> listdlci
```

This displays a list of all DLCIs configured on a particular port.

Expected Results

If you have deleted a particular DLCI, it should not appear on the DLCI list.

Creating an ATM UNI VCI

This section tells you how to create a ATM UNI VCI. Normally, this procedure changes the runtime environment but does not alter ATM UNI VCI information stored on hard disk.

Note To make these changes permanent, issue the command **set config lock** first.

Procedure to Create an ATM UNI VCI

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 Establish the connection by entering the following at the cli> prompt:

```
cli> set port <port#> atm-vci <atm-vci#> <destport> activate
```

where

- <port#> = The frame forwarding port number in card.port format (card = 2 - 10; port = 0 - 1 for ports on an MSC or a CLC).
- <atm-vci#> = The VCI number associated with the ATM UNI connection. The ATM VCI number may be in the range 1-32399, inclusive.
- <destport> = The destination of the ATM VCI. Enter either the IP address or the serial number of the destination switch.

Note The CLI does not display a message indicating that the ATM UNI VCI was created.

Step 3 To verify that the VCI exists, enter the following at the cli> prompt:

```
cli> show port <port#> listvci
```

This displays a list of all VCIs configured on the specified port.

Deleting an ATM UNI VCI

This section tells you how to delete a particular ATM UNI VCI. Deleting an ATM UNI VCI brings down the connection for which this port is an endpoint. Normally, this procedure changes the runtime environment but does not delete the ATM UNI VCI from the configuration information stored permanently on hard disk.

Note To make these changes permanent, issue the command **set config lock** first.

Procedure to Delete an ATM UNI VCI

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the list of DLCIs currently configured on the port, enter the following at the cli> prompt:

```
cli> show port <port#> listvci
```

Step 3 At the cli> prompt, enter the following at the cli> prompt:

```
cli> set port <port number> atm-vci delete
```

where

- <port#> = The port number in card.port format (card = 2 - 10; port = 0 - 1) at one end of the ATM UNI VCI. (You only need to perform the delete action at one end of the DLCI to eliminate it.)
- <atm-vci #> = The number of the ATM UNI VCI you want to delete. The atm-vci number is between 1 - 32399, inclusive.

Note The CLI does not display a message indicating that the ATM UNI VCI was deleted.

Step 4 To verify that the VCI was deleted, enter the following at the cli> prompt:

```
cli> show port <port#> listvci
```

This displays a list of all VCIs configured on a particular port.

Expected Results

If you have deleted a particular VCI, it should not appear on the VCI list.

Setting or Changing Account Passwords

This section tells you how to set or change the password for any user accounts that are defined in your system. To change the passwords for all accounts use the **passwd** command from the LynxOS shell, as described below.

Note When you change the password for the npadmin account, you also change the password for protected mode. (See the alternate procedure for changing passwords in the section “Changing the Protected Mode and npadmin Password.”)

Procedure to Set or Change Passwords

Step 1 To enter protected mode, enter the following at the cli> prompt:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```


Step 3 Use the **shell** command to access the shell to change the passwords, as follows:

```
*cli> shell "passwd <login>"
```

where

<login> =The login name for the account whose password you want to change.

Step 4 Enter the current password when you see the following prompt:

```
Changing password for <login>  
Enter current password:
```

Step 5 Enter the new password when you see the following prompt:

```
Enter new password:
```

The password must be at least six alphanumeric characters long and must not be a recognized name.

Step 6 Retype the new password when you see the following prompt:

```
Retype new password:
```

Expected Results

If you enter the new password correctly, the system changes the password and displays the *cli> prompt.

If you enter an inappropriate password, one or more of the following messages may appear:

```
Please use a longer password.  
Password unchanged.  
Mismatch - password unchanged.
```

Operational Tips

Inform authorized users of the changes you make.

Changing the Default Modem Password and the Modem Initialization String

This section tells you how to change the default modem password and the modem initialization string for the LightStream switch. The modem password and the modem initialization string are stored in EEPROM in the midplane. The default modem password is

```
atmhiway
```

and the default modem initialization string is

```
AT&F&D2&C1&Q0S0=1S2=128S7=30S36=7S95=44
```

You may retain these default values. If you change them, the changes you make are permanent and remain in effect unless you change them again. Rebooting the system or restarting the CLI does not change the modem password or the modem initialization string.

If you change the modem password or the modem initialization string for one switch card slot, make the same change for the other. This is especially important for a two card system because the backup switch card will take over if the active switch card fails. It is also important for a single switch card system because you may want to add an additional switch card later or you may decide to move the single switch card to the other slot.

You must have a switch card in the switch card slot to change the modem password or the modem initialization string. Therefore, if you have only one switch card, move it from one switch card slot to the other as you effect the change for both slots.

Note The modem connected to a LightStream switch must be Hayes-compatible and capable of operating at 2400 baud, which is the only rate at which the LightStream modem port operates.

Procedure to Change the Default Modem Password and the Modem Initialization String

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 To verify that the target switch is correct, enter the following at the *cli> prompt:

```
*cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 4 To change the modem password, enter the following at the *cli> prompt:

```
*cli> set modem <slot #> password <password>
```

where

- <slot #> = The slot number for the switch card (SA or SB) whose modem password you are changing.
- <password> = The new modem password.

Step 5 To change the modem initialization string, enter the following at the *cli> prompt:

```
*cli> set modem <slot #> initstring <initstring>
```

where

- <slot #> = The slot number for the switch card (SA or SB) whose modem initialization string you are changing.
- <initstring> = The new modem initialization string. The format of the modem initialization string should be the same as the default modem initialization string. The content of the modem initialization string depends on the type of modem you are using. Refer to the documentation for your modem to determine the contents of the modem initialization string.

Step 6 To verify the contents of the modem password and the modem initialization string, enter the following at the *cli> prompt:

```
*cli> show modem <slot #> all
```

Expected Results

The password and the modem initialization string are permanently changed. Inform all authorized users of the changes you make.

Operational Tips

Different types of modems require different modem initialization strings. If you have different modems connected to each switch card, the init strings may be different. The passwords may or may not be different.

Inform authorized users of the changes you make.

Changing the Protected Mode and npadmin Password

This section tells you how to change the protected mode password. You can change this password from within protected mode only.

Note When you change the protected mode password, you also change the npadmin password. You can also change the npadmin password with the **passwd** command, as described in the section “Setting or Changing Account Passwords.”

Procedure to Change the Protected Mode and npadmin Password

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

The *cli> prompt appears to indicate that you are in protected mode.

Step 3 At the *cli> prompt, enter:

```
*cli> password
```

Step 4 Enter the protected mode password when you see the following prompt:

```
Changing password for npadmin  
Enter current password:
```

Step 5 Enter the new protected mode password when you see the following prompt:

```
Enter new password:
```

The password must contain at least six alphanumeric characters.

Step 6 Retype the new protected mode password when you see the following prompt:

```
Retype new password:
```

Expected Results

If you retype the new password correctly, the system changes the password and displays the *cli> prompt.

If you enter an inappropriate password, one or more of the following messages may appear:

```
Please use a longer password.  
Password unchanged.  
Please use a less obvious password.  
Passwords don't match, try again.
```

Operational Tips

Note Inform all authorized users of the changes you make.

Setting the SNMP Community

Each SNMP manager (the CLI, for example) and each managed system (the MMA in a LightStream switch, for example) has a community name. The SNMP manager specifies a community name in each command it sends. The managed system validates the commands before executing them by comparing the community name in the command against its own community name.

Before you can set attributes or use CLI control commands, you must set the SNMP community to a community that has read/write access privileges. The read/write community provided with the system is named *write*. (A switch can have several SNMP community names with read/write privileges.) The read-only community provided with your system is named *public*.

To prevent unauthorized access to your system, you should set the SNMP community names that the LightStream switch uses to validate the commands before it executes them. (Procedures to change the read/write community name are described in the section “Changing the Default SNMP Community Names.”) Follow the procedure below to set the SNMP community name that the CLI puts in commands.

Procedure to Set the SNMP Community Name

Step 1 At the cli> prompt, enter:

```
cli> set snmp community <community name>
```

where

<community name> = The name for the SNMP community with read/write privileges that you want to access.

Step 2 To verify the SNMP community name, enter the following at the cli> prompt:

```
cli> show snmp
```

Expected Results

The community name is set to the SNMP community you specified.

Operational Tips

The SNMP community reverts to the read-only community when you log out of the CLI. However, if you leave your terminal without logging out of the CLI, be sure to change the SNMP community back to the read-only community to prevent unauthorized access to your system.

Creating the cli.groups File

The cli.groups file defines groups of traps. You can use this file as an argument for the commands described in the “Using LightStream Traps” chapter. If you do not create and maintain this file, you must manually enter each trap number used with those commands.

Follow the procedure below to create the cli.groups file.

Procedure to Create the cli.groups File

Step 1 To enter protected mode, enter the following at the cli> prompt:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Escape from the CLI to the bash shell, enter the following at the *cli> prompt:

```
*cli> shell bash
```

Step 4 Move to the /usr/app/base/config directory by entering the following at the bash# prompt:

```
bash# cd /usr/app/base/config
```

Step 5 Invoke the vi editor by entering the following at the bash# prompt:

```
bash# vi cli.groups
```

When the editor opens the file, enter the group names and trap numbers in the format shown below.


```
<group# name> <trap#> <trap#>
<group# name> <trap#> <trap#>
```

where

- <group# name> = A name that defines the group of traps.
- <trap#> = The trap numbers within the group.

Your file will be similar to the file shown in Figure 3-2.

Figure 3-2 Sample cli.groups file



```
bash# vi cli.groups
group1 name> <trap#> <trap#>
```

Step 6 When you have finished entering the group names and trap numbers in the file, exit the vi editor by pressing the [Esc] key or ^[followed by typing:

```
ZZ
```

Step 7 To return to the CLI, enter the following at the bash# prompt:

```
bash# exit
```

Step 8 To exit protected mode, enter the following at the *cli> prompt:

```
*cli> exit
```

Creating CLI Script Files

This section tells you how to create CLI script files that can be executed from the CLI. You can write CLI script files to perform a variety of functions and reduce the number of commands required to perform repetitive tasks. The CLI script files are placed in the working directory. For example, if you write the CLI script file from the oper account, the CLI script files are placed in /usr/oper directory.

Procedure to Create CLI Script Files

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Escape from the CLI to the bash shell, enter the following at the *cli> prompt:

```
*cli> shell bash
```

Step 4 Invoke the vi editor by entering the following at the bash# prompt:

```
bash# vi <file name>
```

where

<file name> = The name of the file containing the CLI script.

Step 5 Use the vi editor to enter a series of CLI commands into your CLI script file. If you are unfamiliar with the vi editor, refer to the *LightStream 2020 Command and Attribute Reference Guide*.

A sample CLI script file might consist of the following lines:

```
show chassis general
show chassis cards
show card 1 ports
show card 2 ports
show card 3 ports
show card 4 ports
show card 5 ports
show card 6 ports
show card 7 ports
show card 8 ports
show card 9 ports
show card 10 ports
```

This file displays chassis information and lists all boards and ports in the chassis. If no card exists in a slot, the system reports that condition and continues with the next line in the file.

Step 6 When you have entered all lines into the CLI script file, exit the vi editor by pressing the [Esc] key or ^[followed by typing:

```
ZZ
```

Step 7 To return to the CLI, enter the following at the `bash#` prompt:

```
bash# exit
```

Step 8 To exit protected mode, enter the following at the `*cli>` prompt:

```
*cli> exit
```

You can now run the CLI script file using the **source** command. Refer to the *LightStream 2020 Operations Guide* for instructions on how to run a CLI script file.

Performing an Orderly Shutdown

This section tells you how to power down your switch to ensure that the flow of data through the switch terminates gracefully. Three procedures are provided: the first is for shutting down a switch with two NPs; the second is for shutting down a switch with one NP; the third is for returning a switch to service.

Procedure 1: Shutting Down a Switch with Two NPs

In a system with two NPs, you must reboot the backup NP *before* you reboot the active NP. (If you reboot the active NP first, the backup takes over and the system continues to operate.)

Step 1 Warn anyone who will be affected that you're taking the system out of service.

Step 2 Log in to the root account on the switch you want to shut down.

Step 3 To determine which NP is active (primary), start the CLI and use the command **show chassis general** and look for Slot of Primary NP in the resulting display. This is the NP you'll reboot *last*.

Step 4 To log in to the backup NP (the one whose slot number was *not* displayed in Step 3), do the following:

- Type `\.` to get a TCS hub prompt.
- At the TCS hub prompt, use **connect <slot#>** to connect to the backup NP. The example below assumes that you are connecting to the NP in slot 2.

```
TCS hub<<A>> connect 2
```

- Log in to the root account.

Step 5 From the `bash#` prompt, type

```
bash# reboot -n
```

Step 6 Type `\.` to return to the TCS hub.

Step 7 At the TCS hub prompt, use **connect <slot#>** to connect to the active NP. The example below assumes that you are connecting to the NP in slot 1.

```
TCS hub<<A>> connect 1
```

Step 8 If necessary, type **quit** to exit from CLI and get a `bash#` prompt.

Step 9 From the `bash#` prompt, type

```
bash# reboot -n
```

Step 10 You can now turn off the power.

Procedure 2: Shutting Down a Switch with One NP

Step 1 Warn anyone who will be affected that you're taking the system out of service.

Step 2 Log in to the root account on the switch you want to shut down.

Step 3 From the bash# prompt, type

```
bash# reboot -n
```

Step 4 You can now turn off the power.

Procedure 3: Returning to Service

If you turned off the power, you can return the system to service by turning the power on again.

If you did not turn off the power, use the command shown below at the TCS hub to bring the system back to service. Replace *slot#* with the slot number of the NP card (1 or 2):

```
TCS hub<<A>> reset <slot#>
```

Note that you must issue the **reset** command to both NPs in a redundant system.

Handling Cell Line Card Interfaces

The LightStream switch supports the cell line card (CLC).

Displaying CLC Attributes

Use the CLI **show** command as described in this procedure to display information about the CLC.

Procedure to Display CLC Attributes

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To display the status of a CLC port, enter the following at the cli> prompt:

```
cli> show port <port#> status
```

where

<port#> = The card and port number of the CLC in card.port format (card = 2 - 10; port = 0 or 0 - 1).

Note The CLI no longer uses the long port format designation (for example, 900,4002, etc.). You must use the dotted format (for example, 9.0, 4.2)..

The displayed screen will be similar to the following:

```
cli> show port 5.0 status
```

```
Description: CLSC1 OC3 Trunk Line Card Rev
```



```
1.0Port
Name: tb3.5.0->5:2,0
Port Type: OC3 Trunk
MIB2 Type: sonet
Port MTU: 53 Octets
Port Speed: 155520000 bps

Admin Status: Up
Oper Status: Up
Oper loop: none
Admin loop: none
Last Oper Change: 34 Min 30 Sec ago

Octets Rcvd: 8810826
Normal Packets Rcvd: 166242
Multicast Packets Rcvd: 0
Discarded Rcvd Packets: 60
Receive Errors: 74
Unknown Protocols Rcvd: 166242
Octets Sent: 2272216
Normal Packets Sent: 42873
Multicast Packets Sent: 0
Discarded Output Packets: 0
Output Errors: 140

Oper Protocol: CLC Trunk
Admin Protocol: CLC Trunk
Port Data Cell Capacity: 349674 cells
Port Unreserved Capacity: 34674 cells
Link Transmit Utilization: 12 cells/sec.
Clocking: internal
Medium
Time Elapsed in current interval: 586
Number of valid interval: 2
Section
Status: <No defects>
Error seconds: 0
Severe error seconds: 0
Severe framing error seconds: 0
Coding violations: 0
Line
Status: <No defects>
Error seconds: 0
Severe error seconds: 0
Unavailable seconds: 0
Coding violations: 0
Path
Path width: sts3cSTM1
Status: <No defects>
Error seconds: 0
Severe error seconds: 0
Unavailable seconds: 0
Coding violations: 0
cli>
```

Step 3 To display the physical state of a CLC,

```
cli> show port <port#> physical
```

A screen similar to the following will be displayed:

```
cli> show port 4.0 physical

Oper Protocol: CLC Trunk
AdminProtocol: CLC Trunk
```

```
Port Data Cell Capacity: 349674 cells
Port Unreserved Capacity 349674 cells
Link Transmit Utilization: 3 cells/sec.
Clocking: internal
```

```
cli>
```

Expected Results

Loop status (internal, external, remote, and none) are displayed for the CLC board.

Handling FDDI Interfaces

This section describes the LightStream CLI commands used to support the FDDI interfaces.

Specifying FDDI Attributes

Use the CLI **show** and **set** commands as described in this procedure to display and specify FDDI information.

Procedure to Specify FDDI Attributes

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To determine which cards are FDDI cards, enter the following commands at the cli> prompt:

```
cli> show chassis cards
```

A screen similar to the following will be displayed:

```
cli> show chassis cards
```

```
Slot 1: NP
Slot 2: Empty
Slot :3 LS Edge
Slot :4 LS Trunk
Slot :5 MS Trunk
Slot :6 ATM-UNI
Slot :7 MS Trunk
Slot :8 LS Trunk
Slot :9 FDDI
Slot :10 LS Edge
Slot :SA Switch
Slot :SB Empty
cli>
```

Step 3 To enable or disable a FDDI port, enter the following commands at the cli> prompt:

```
cli> set port <port#> fddi {aport|bport|smt} <action>
```

where

- <port#> = The card and port number of the FDDI card in card.port format (card = 2 - 10; port = 0 - 2).
- <action> = enable, disable, start, or stop.

For example, to enable FDDI port 9.1, you would enter the following command:

```
cli> set port 9.1 fddi aport action enable
```

To set FDDI port 9.1 station management parameters, setting the Neighbor Notification protocol timer to 20 seconds, and stop sending Status Reporting Frames for FDDI events to the SMT management software, you would enter the following command:

```
cli> set port 9.1 fddi smt t-notify 20 stat-report no
```

Step 4 To display FDDI details for the port, enter the following at the cli> prompt:

```
cli> show port <port#> fddi smt
```

A screen similar to the following is displayed:

```
cli> show port 9.1 fddi smt
SMT Index: 8
Station Id: 0 : 0 : 10 : 0 : 10 : 0 : 14 : ff
Operation Version: 2
Highest SMT Version: 2
Lowest SMT Version: 2
User Data: 831016
FDDI MIB Version: 1
Number of MACs: 1
Number of A, B or S ports: 2
Number of M ports: 0
Available PATH types: Primary, Secondary,
Configuration Capabilities: hold<notsupported>,CF-Wrap<notsupported>
Configuration Policy: ConfigurationHold<inactive>
Connection Policy: 34629
Neighbor Notification Tim 20 seconds
Generate Status Reporting no
Trace Max Expiration: Unknown msec.
Bypass on AB port: False
ECM State: In
TCF State: isolated
Remote Disconnect Flag: False
Station Status: concatenated
Peer Wrap Flag: False
Time Stamp: 2643795 msec.
Transition Time Stamp: 2758715 msec.
Station Action: other
cli>
```

Expected Results

The FDDI parameters are set as you specified.

Handling Ethernet Interfaces

This section tells you the LightStream CLI commands used to support Ethernet interfaces.

Specifying Ethernet Information

Use the CLI **show** and **set** commands as described in this procedure to display and specify Ethernet information.

Procedure to Specify Ethernet Information

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To set an Ethernet interface to active, enter the following command at the cli> prompt:

```
cli> set <port#> active  
<action> {active|inactive}
```

where

<port#> = The card and port number of the Ethernet card in card.port format (card = 2 - 10; port = 0 - 7).

Step 3 To display the status of an Ethernet card, enter the following at the cli> prompt:

```
cli> show <port#>
```

A screen similar to the following is displayed:

```
cli> show port 3.0  
  
Description: Ethernet CSMACD  
Port name:  
Port Type: Ethernet  
MIB2 Type: ethernet-csmacd  
Port MTU: 1500 Octets  
Port Speed: 10000000 bps  
Admin Status: Up  
Oper Status: Up  
Last Oper Change: 1 Hr 24 Min 5 Sec ago  
Octets Rcvd: 0  
Normal Packets Rcvd: 0  
Multicast Packets Rcvd: 0  
Discarded Rcvd Packets: 0  
Receive Errors: 0  
Unknown Protocols Rcvd: 0  
Octets Sent: 169280  
Normal Packets Sent: 0  
Multicast Packets Sent: 2646  
Discarded Output Packets: 0  
Output Errors: 0  
  
Alignment Errors: 0  
FCS Errors: 0  
Single Collision Frames: 0  
Multiple Collision Frames: 0  
SQE Test Errors: 0  
Deferred Transmissions: 0  
Late Collisions: 0  
Excessive Collisions: 0  
Internal Mac Transmit Errors: 0  
Carrier Sense Errors: 0  
Excessive Deferrals: 0  
Frame Too Longs: 0
```

```
In Range Late Errors: 0
Out of Range Length Fields: 0
Internal Mac Receive Errors: 0

Card Port WgrpId Mode
3 0 1 Include
Default action is FORWARD
```

Handling Spanning Tree Bridging

The LightStream switch uses the Spanning Tree Protocol to detect loops within a bridged network. When a loop is detected, one port on the bridge performs a blocking function to break the loop. All bridging traffic on that port is discarded and MAC address learning is not performed. This section tells you the LightStream CLI commands used to support spanning tree bridging.

Use the CLI **show** and **set** commands as described in this section to display and specify spanning tree bridging information.

Defining Spanning Tree Bridge Parameters

This procedure describes how to define and display spanning tree bridge parameters.

Procedure to Define Spanning Tree Bridge Parameters

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

Step 2 To view the current general spanning tree bridge parameters, enter the following at the cli> prompt:

```
cli> show stb general
```

A screen similar to Figure 3-3 is displayed.

Figure 3-3 Sample stb general display

```
cli> show stb general

Bridge Max Age:      2000
Bridge Hello Timer:  200
Bridge Forward Delay: 1500
Priority:             0

cli>
```

H33355

Step 3 To set the spanning tree timeout parameters, enter the following commands at the cli> prompt:

```
cli> set stb maxage <maxagevalue>
```

where

<maxagevalue> = The maximum interval that is used to time out spanning tree information

```
cli> set stb forwdelay <fwd-delay-val>
```

where

<fwd-delay-val> = The time interval to be used before changing to another state

```
cli> set stb hellotimer <hello-timer-val>
```

where

<hello-timer-val> = The time interval between Hello BPDUs

```
cli> set stb priority <priority>
```

where

<priority> = The priority for using this node vs. others for a path using the Spanning Tree Protocol. The range is 0 - 65535, and the default is 32768.

Step 4 To verify that the spanning tree parameter changes have been made, enter the following at the cli> prompt:

```
cli> show stb general
```

A screen similar to Figure 3-3 is displayed. Your changes should appear in the display.

Expected Results

The spanning tree parameters are set as you specified.

Defining Spanning Tree Static Filters

This section tells you how to make entries into the bridge filtering database.

Procedure to Define Spanning Tree Static Filters

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the current statically entered spanning tree bridge filtering entries, enter the following at the cli> prompt:

```
cli> show stb static
```

Step 3 To make entries into the spanning tree bridge static filtering database, enter the following at the cli> prompt:

```
cli> set stb static <MACaddr> rcv <rcv-port> xmit <xmit-port>
```

where

- <MACaddr> = The is the MAC address. The MAC address must be entered in xx:xx:xx:xx:xx:xx format.
- <rcv-port> = The port to which this MAC address is assigned. This may be a number or the keyword **any**.

- <xmit-port> = The port to which received frames are to be forwarded.

Step 4 To verify that your entries have been made, enter the following at the cli> prompt:

```
cli> show stb static
```

Your entries should appear in the display.

Step 5 To view the spanning tree bridge forwarding table entries and their associated variables, enter the following at the cli> prompt:

```
cli> show stb fwd
```

Your entries should appear in the display.

Step 6 To view bridge port information, enter the following at the cli> prompt:

```
cli> show stb ports
```

Handling Bridge Filters

LightStream custom filtering allows you to define filters to block or forward incoming frames for specific ports. You must first define the bridge filter and then associate the filter with a port or ports.

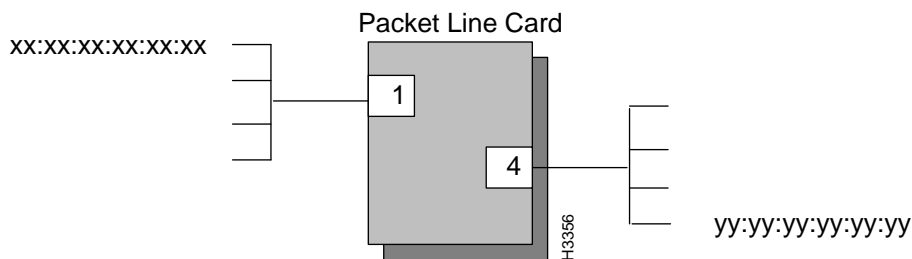
The filter is a set of conditions that is compared to information in the header of incoming frames. As an incoming packet is received, its level 2 header is broken into components. The header information is evaluated (in priority order) against all filters associated with the receiving port. If a filter condition matches the header information, the action specified by that filter is taken. If the filter condition does not match the frame header information, the next filter is evaluated. If no filter conditions match the frame header information, the default action for the port is taken.

Defining Bridge Filters

The procedure in this section describes how to define a custom bridge filter. To define a bridge filter, you first assign a number to the filter and then you write the filter expression. For a description of filter attributes, construction, and examples, refer to the *LightStream 2020 Command and Attribute Reference Guide*.

Procedure to Define Bridge Filters

This procedure defines sample bridge filters that will block the LAN end stations in Figure 3-4 from communicating with each other. To successfully block the communications, filters must be created for the ports (1 and 4) supporting each LAN.

Figure 3-4 Connections to be filtered

Step 1 To determine if any filters are currently defined for either of these ports, enter the following at the cli> prompt:

```
cli> show <port#> bflt
```

where

<port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).

Step 2 To display a current filter, enter the following at the cli> prompt:

```
cli> show bflt <bflt-id>
```

where

<bflt-id> = The number that identifies the filter whose contents are to be displayed.

Step 3 To define the new filter for port 1, blocking all traffic from the source end station (xx:xx:xx:xx:xx:xx) that is directed to the destination end station (yy:yy:yy:yy:yy:yy), enter the following at the cli> prompt:

```
cli> define bflt <bflt-id> macSrc == xx:xx:xx:xx:xx:xx && macDst == yy:yy:yy:yy:yy:yy
```

where

<bflt-id> = The identifying number that you assign to the filter.

Step 4 To define the new filter for port 4, blocking all traffic from the source end station (yy:yy:yy:yy:yy:yy) that is directed to the destination end station (xx:xx:xx:xx:xx:xx), enter the following at the cli> prompt:

```
cli> define bflt <bflt-id> macDst == xx:xx:xx:xx:xx:xx && macDst yy:yy:yy:yy:yy:yy
```

You must now assign each filter to the appropriate ports.

Step 5 To associate the appropriate filter with port 1, enter the following at the cli> prompt:

```
cli> set port <port#> bflt <bflt-id> <action> <priority>
```

where

- <bflt-id> = The identifying number that you assigned to the filter created in Step 3.
- <action> = The action that is to be taken when the frame value matches the filter value (block, in this case).

- **<priority>** = The priority number applied to this filter. The filter is added to a priority list according to this value. Incoming frames are compared to the filters in priority order. One is the highest priority. It is recommended that you assign the priorities by 10s (10, 20, 30, and so forth) to leave ample numbers available for the reordering or adding priorities.

Step 6 To associate the appropriate filter with port 4, enter the following at the cli> prompt:

```
cli> set port <port#> bflt <bflt-id> <action> <priority>
```

where

<bflt-id> = The identifying number that you assigned to the filter created in Step 3.

If the default for both of these ports is to forward, then all other traffic is allowed (unless, of course, other filters have been defined to block certain traffic).

Step 7 To verify that the filters have been defined, enter the following at the cli> prompt:

```
cli> show bflt
```

Expected Results

The filters you created now block traffic from being sent between the end stations on the LANs.

Assigning a Filter to a Specific Port

Any filter can be assigned to any port at any time. Incoming frames for that port are subsequently compared with the filter conditions. If the value of a specific field in the frame header matches the value of the filter, the action specified by the filter condition is taken. Follow the procedure below to associate a filter with a specific port or ports.

Procedure to Assign a Filter to a Specific Port

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the filters currently defined for a specific port, enter the following at the cli> prompt:

```
cli> show <port#> bflt
```

where

<port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).

Step 3 To associate the filter with a specific port, enter the following at the cli> prompt:

```
cli> set port <port#> bflt <bflt-id> <action> <priority>
```

where

- **<bflt-id>** = The number that identifies the filter being assigned to the port.
- **<action>** = The action that is taken when the frame value matches the filter value (forward or block).

- <priority> = The priority number applied to this filter. The filter is added to a priority list according to this value. Incoming frames are compared to the filters in priority order. One is the highest priority. It is recommended that you assign the priorities by 10s (10, 20, 30, and so forth) to leave ample numbers available for the reordering or adding priorities.

Note The maximum number of filters that can be assigned to a port is 32. The maximum number of filters that can be assigned to all ports is 1024.

Defining the Default Filter Action

This procedure describes how to define the default filter action for a specific port. This determines the action to take with incoming traffic (forward or block) when incoming traffic matches none of the defined filter conditions.

Procedure to Define the Default Filter Action

- Step 1** To view the current default filter action parameter for a specific port, enter the following at the cli> prompt:

```
cli> show port <port#> bflt-def
```

where

<port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).

- Step 2** To define or alter the default filter action parameter for this port, enter the following at the cli> prompt:

```
cli> set port <port#> bflt-def
```

where

<def-action> = The default action (either forward or block) for this port.

- Step 3** To verify the change, enter the following at the cli> prompt:

```
cli> show port <port#> bflt-def
```

Defining the Default Broadcast Limit

This procedure describes how to define the default broadcast limit parameter for a specific port.

Procedure to Define the Default Broadcast Limit

- Step 1** To view the current default broadcast limit parameter for a specific port, enter the following at the cli> prompt:

```
cli> show port <port#> bcast-limit
```

where

<port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).

- Step 2** To define or alter the default broadcast limit parameter for this port, enter the following at the cli> prompt:

```
cli> set port <port#> <bcast-limit>
```

where

- <def-action> = The default action (either forward or block) for this port.
- <bcast-limit> = A range of 1 to 127 (the number of broadcast packets to forward per second); discard all (discard all broadcast packets); or forward all (forward all broadcast packets).

Step 3 To view the default broadcast limit parameter, enter the following at the cli> prompt:

```
cli> show port <port#> bcast-limit
```

Step 4 To define or alter default broadcast limit for this port, enter the following at the cli> prompt:

```
cli> set port <port#> bcast-limit <bcast-limit>
```

where

<bcast-limit> = The maximum number of broadcast frames per second that is forwarded. The excess broadcast frames are dropped. The maximum number of broadcast frames per second is 128. To block all frames from being forwarded, assign 0 to this parameter. To forward all frames, assign -1 to this parameter.

Removing a Filter Associated with a Port

This procedure tells you how to disassociate a filter from a specific port or ports.

Procedure to Remove a Filter from a Port

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the filters currently defined for a specific port, enter the following at the cli> prompt:

```
cli> show <port#> bflt
```

Step 3 To break the association between a filter and a port, enter the following at the cli> prompt:

```
cli> set port <port#> bflt <bflt-id> delete
```

where

- <port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).
- <bflt-id> = The number that identifies the filter being assigned to the port.

This command removes the specified filter from the list of filters associated with a port. The filter is still defined but does not affect traffic on the specified port.

Step 4 To verify that the association was removed, enter the following at the cli> prompt:

```
cli> show <port#> bflt
```

Deleting a Filter

This procedure describes how to delete a filter. You cannot delete a filter that is associated with a port. You must first perform the procedure described in the section “Removing a Filter Associated with a Port.”)

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the currently defined filters, enter the following at the cli> prompt:

```
cli> show bflt
```

Step 3 To view the filters currently defined for a specific port, enter the following at the cli> prompt:

```
cli> show <port#> bflt
```

where

<port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).

Step 4 To delete a filter, enter the following at the cli> prompt:

```
cli> delete bflt <bflt-id>
```

where

<bflt-id> = The number that identifies the filter.

Note If the filter is associated with a port, the delete action is rejected. Perform the procedure Removing a Filter Associated with a Port, then delete the filter.

Step 5 To verify that the filter was deleted, enter the following at the cli> prompt:

```
cli> show bflt
```

The filter you deleted should not appear in the display.

Handling Virtual LAN Internetworking

Virtual LAN Internetworking (VLI) allows you to transcend the physical limitations of LAN internetworking. The LightStream configurator allows you to arrange stations in distinct workgroups and to restrict access between workgroups. Stations on different physical segments can belong to the same workgroup, and they can belong to more than one workgroup. For further information, refer to the *LightStream 2020 Configuration Guide*.

Establishing the Default Workgroup

This procedure describes how to establish the default workgroup. The default workgroup is established by having no workgroup IDs at all in an exclude list; that is, excluding no one. An exclude list that is not empty includes everybody except those that have at least one of the listed workgroup IDs in their include list. An include list admits *only* those that have at least one of the listed workgroup IDs on their include list. An empty include list blocks all communications.

Procedure to Establish Lists

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To create an include list, enter the following at the cli> prompt:

```
cli> set port <port#> wgrp include
```

where

<port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).

Step 3 To create an exclude list, enter the following at the cli> prompt:

```
cli> set port <port#> wgrp exclude
```

Adding a Workgroup to a Port List

This procedure describes how to add a workgroup ID to a list for a specific port.

Procedure

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To add a workgroup to the list for the port, enter the following at the cli> prompt:

```
cli> set port <port#> wgrp add <wgrp#>
```

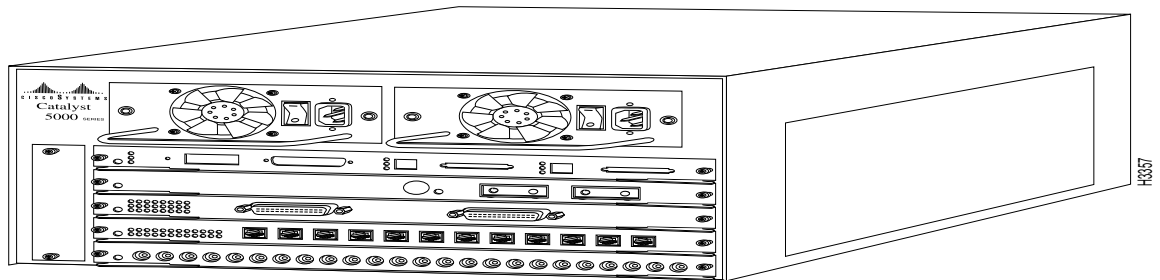
where

- <port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).
- <wgrp#> = The number that identifies the workgroup.

Step 3 To verify that the workgroup was added to the list, enter the following at the cli> prompt:

```
cli> show port <port#> wgrp
```

A screen similar to Figure 3-5 is displayed.

Figure 3-5 Example - show port wgrp display

Removing a Workgroup from a Port List

This procedure describes how to delete a workgroup ID from a list for a specific port.

Procedure

Step 1 To verify that the target switch is correct, enter the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the *LightStream 2020 Operations Guide*.

Step 2 To view the workgroups currently defined for a specific port, enter the following at the cli> prompt:

```
cli> show port <port#> wgrp
```

where

- <port#> = The card and port number in card.port format (card = 2 - 10; port = 0 - 7).
- <wgrp#> = The number that identifies the workgroup.

Step 3 To disassociate the workgroup from a port, enter the following at the cli> prompt:

```
cli> set port <port#> wgrp delete <wgrp#>
```

Step 4 To verify that the association was removed, enter the following at the cli> prompt:

```
cli> show port <port#> wgrp
```

A screen similar to Figure 3-6 is displayed.

Figure 3-6 Example - show port wgrp display

```
cli> show port 5.7 wgrp
```

<u>Workgroup List</u>			
<u>Card</u>	<u>Port</u>	<u>WgrpId</u>	<u>Mode</u>
5	7	1	Exclude

```
cli>
```

H3358

SNMP Commands

SNMP Procedures • SNMP Command Arguments

This chapter describes the SNMP commands available from the command line interface (CLI) on a LightStream 2020 enterprise ATM switch. A section is included describing how to identify the management information base (MIB) address used with the commands. This chapter is intended for SNMP experts who will use the commands to monitor and manipulate MIB objects. If you are not familiar with SNMP, it is not recommended that you use these commands.

In some cases, SNMP commands may be long and repetitive. You can store sequences of SNMP commands in CLI script files and execute them using the **source** command described in the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

SNMP Procedures

The CLI provides four SNMP commands:

- getsnmp
- getnextsnmp
- setsnmp
- walksnmp

The first three commands are equivalent to the standard SNMP commands **getrequest**, **getnextrequest**, and **setrequest**. The fourth command allows you to display MIB objects.

SNMP Command Arguments

MIB Object Names

MIB *object names* are used to identify the MIB objects in the SNMP commands described in this chapter. MIB object names come from the standard or private MIBs that the LightStream switch supports. Refer to the “Getting Started” chapter of the *LightStream 2020 Command and Attribute Reference Guide* for a detailed discussion of the MIB.

MIB Instance Identifier

In addition to a MIB object name, you must use a MIB *instance identifier* to manipulate a particular MIB object. SNMP calculates a suffix based on the hierarchy of a MIB object and adds the suffix to the MIB object name to form its instance identifier. This use of instance identifiers allows all MIB object names to be unique.

For example, the MIB object *sysDescr* has only a single value. It is identified by its MIB object name (*sysDescr*) followed by its instance identifier (0), resulting in *sysDescr.0*.

A MIB object with multiple instances has a unique identifier for each instance. In the following examples, the number (1, 16, 24, or 43) following the MIB object *pidName* identifies the specific instance of *pidName*:

- pidName.1
- pidName.16
- pidName.24
- pidName.43

For a further discussion of instance identifiers, refer to *The Simple Book: An Introduction to Management of TCP/IP-based Internets* by Marshall T. Rose, 1991, Prentice Hall, Inc. (ISBN 0-13-812611-9).

Identifying MIB Objects

If you are unsure of a MIB object name, use the **walksnmp** command with the specific object's MIB tree or subtree name as the command argument. (See the section, "Walking a MIB Subtree.") This displays all MIB objects below the specified point in the tree. Whenever the attribute MIB name or MIB address is required by a syntax statement, you must enter both the MIB object name and its instance identifier.

Monitoring the Value of a MIB Object

The procedures in this section tell you how to view the value of a particular MIB object using one of two different commands. Procedure 1 using the **getsnmp** command displays the value of the MIB object you specify. Procedure 2 using the **getnextsnmp** command displays the value of the MIB object following the MIB object you specify. The **getnextsnmp** command is useful if you don't know the exact structure of the MIB.

Procedure 1: Displaying the Value of a Specified MIB Object

At the cli> prompt, enter:

```
cli> getsnmp <MIB name or address>
```

where

<MIB name or address> = Name or address of the MIB object you want information on. You may enter multiple names or addresses.

Some examples of the getsnmp command are

```
getsnmp sysDescr.0
getsnmp sysDescr.0 sysObjectID.0 sysUpTime.0
getsnmp 1.3.6.1.2.1.1
```

Procedure 2: Displaying the Value of the MIB Object After a Specified MIB Object

At the cli> prompt, enter:

```
cli> getnextsnmp <MIB name or address>
```

where

<MIB name or address> = Name or address of the MIB object just before the MIB object you want to get information on. You may enter multiple names or addresses.

Expected Results

When you enter **getsnmp sysDescr.0**, the following information is displayed:

```
cli> getsnmp sysDescr.0
Name: sysDescr.0 Value: ATM Data Switch
cli>
```

When you enter **getnextsnmp sysDescr.0**, the following information is displayed:

```
cli> getnextsnmp sysDescr.0
Name: sysObjectID.0 Value: SWITCH_SNMP_Agent
cli>
```

Setting a MIB Object

This procedure tells you how to set the value of a MIB object using the **setsnmp** command. If you change a MIB object using this command, the change is not saved permanently to hard disk. Instead, the changes are made to runtime memory. If you reboot the system, the changes made using this procedure are lost.

Procedure

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Set the SNMP community to a read/write community by entering the following at the *cli> prompt:

```
*cli> set snmp community <community name>
```

where

<community name> = The name of the SNMP community with read/write privileges that you want to access. (A switch can have several SNMP community names with read/write privileges.)

Note The SNMP community must be set to a read/write community before you can set the MIB object value.

Step 4 At the *cli> prompt, enter:

```
*cli> setsnmp <MIB name or address> <value>
```

where

- <MIB name or address> = Name or address of the MIB object you want to change. You may enter multiple names or addresses and their corresponding values.

- <value> = the value of the MIB object you want to set.

For example, you can enter

```
setsnmp chassisId 4143
```

Expected Results

Whenever you issue a **setsnmp** command from CLI, it is automatically followed by a **getsnmp** command. Refer to the information displayed by the **getsnmp** command to verify that the change you made is correct. After entering the command shown above, the following would be displayed:

```
Name: chassisId.0      Value: 4143
```

Walking a MIB Subtree

This procedure tells you how to display a portion of the MIB subtree. Using the **walksnmp** command, you can enter the name of the subtree you want to “walk” through. This command then displays all MIB objects (and their values) in that subtree. You can also use **walksnmp** to get a list of all the instances of a particular MIB object in the switch.

Procedure

At the cli> prompt, enter:

```
cli> walksnmp <MIB name or address>
```

where

<MIB name or address> = The name or address of a MIB subtree or MIB object. (You must put the name in quotes if it contains non-alphanumeric characters)

For example, to walk the System subtree, enter:

```
cli> walksnmp system
```

or

```
cli> walk mib2
```

or

```
cli> walksnmp 1.3.6.1.2.1.1
```

To list the names of all the processes running in the LightStream switch, enter:

```
cli> walksnmp pidName
```

Expected Results

When you enter **walksnmp system**, the information shown in Figure 4-1 is displayed:

Figure 4-1 Example of walksnmp system display

```
cli> walksnmp system
Name: sysDescr.0      Value: ATM Data Switch
Name: sysObjectID.0   Value: SWITCH_SNMP_Agent
Name: sysUpTime.0     Value: 39501284
Name: sysContact.0    Value: Tom Matthews
Name: sysName.0       Value: emtb7
Name: sysLocation.0   Value: Cambridge
Name: sysServices.0   Value: 78
cli>
```

H3320

When you enter **walksnmp pidname**, the information shown in Figure 4-2 is displayed:

Figure 4-2 Example of walksnmp pidname

```
cli> walksnmp pidName

Name: pidName.8      Value: ndd
Name: pidName.9      Value: watchdog
Name: pidName.10     Value: mma
Name: pidName.16     Value: kernlog
Name: pidName.17     Value: cac
Name: pidName.18     Value: collector
Name: pidName.19     Value: nptmm
Name: pidName.20     Value: npcc
Name: pidName.21     Value: gidd
Name: pidName.23     Value: lcc
Name: pidName.24     Value: lcc
Name: pidName.25     Value: lcc
Name: pidName.33     Value: lcc
cli>
```

H3321

TCS Hub Commands

Access to the TCS Hub Interface • User Interface for TCS Hub Commands • TCS Hub Tasks

This chapter tells you how to access the Test and Control System (TCS) interface and how to use TCS hub commands to manage your network of LightStream 2020 enterprise ATM switches. TCS hub commands allow you to communicate directly to the TCS and they allow you to perform some functions when you do not have access to the command line interface (CLI). You can use the TCS hub commands to do the following:

- Get help on any TCS hub command
- Connect to a card
- Force a TCS hub on either of the switch cards (SA or SB) to become the primary or secondary TCS hub

Each of these tasks is described in this chapter. The commands discussed here are a subset of the TCS hub commands. The remaining TCS hub commands are not described here because you can use CLI commands to perform the same functions. Refer to the *LightStream 2020 System Overview* for a further discussion of the TCS.

Access to the TCS Hub Interface

You access the TCS hub interface by connecting a VT100-compatible terminal (or modem) to the TCS console (or modem) port on the back of the LightStream switch. Press the [Return] key to obtain a TCS prompt.

Prompts

When you access the TCS, you see one of a variety of prompts. Table 5-1 explains each of the possible prompts.

Table 5-1 TCS Prompt Descriptions

Prompt	Character	Case
TCS HUB<A> tcs hub<a> TCS HUB<<A>> tcs hub<<a>>	The character A or a indicates you are connected to the TCS hub on switch card A.	A prompt shown in uppercase characters (TCS HUB<A>) indicates that the TCS hub is the primary TCS hub. A prompt in lowercase characters indicates the TCS hub is the secondary TCS hub.
TCS HUB tcs hub TCS HUB<> tcs hub<> ¹	The character B or b indicates you are connected to the TCS hub on switch card B.	A prompt shown in uppercase characters (TCS HUB) indicates that the TCS hub is the primary TCS hub. A prompt in lowercase characters indicates the TCS hub is the secondary TCS hub.

1. The significance of the single or double brackets in the prompt is not relevant to this discussion.

User Interface for TCS Hub Commands

The TCS interface is similar to that of the CLI. Each CLI command includes the object identifier of the component to which the command is directed (chassis, card, port, etc.). A TCS hub command is always directed to a card, so it includes the slot number rather than the object identifier.

Online Help

Like the CLI, the TCS interface has an online help facility. It provides a **help** command, and it also allows you to enter a question mark (?) to list options for a command. Unlike CLI, the TCS does not provide for a command completion feature using the [Tab] key.

Line Editing Keys

The TCS provides a number of line editing keys. Some of these keys are different from those used in the CLI. Table 5-2 lists the TCS line editing keys.

Table 5-2 TCS Command Line Editing Keys

Control Keys	Function
^P	Yank previous command.
^N	Yank next command.
^A	Go to start of line.
^E	Go to end of line.
^B	Go back one character.
^F	Go forward one character.
^U	Kill entire line.
^D	Delete character under cursor.
[Esc] B	Go back one word.
[Esc] F	Go forward one word.

Getting Help on TCS Commands

This procedure tells you how to access online help on any TCS hub command. When you enter the **help** command, the TCS displays a list of all commands available as shown in Figure 5-1. When you enter the **help** command followed by a command name, the system displays information on that command as shown in Figure 5-2.

Procedure

At any TCS prompt, enter:

```
TCS HUB<A>> help [<command>]
```

where

[<command>] = An optional argument. The name of any TCS hub command on which you want help.

Figure 5-1 TCS help screen

```

TCS HUB<<A>> help

    baud          <rate>
    connect       <slot>
    help          <command>
    init          <slot><init_string>
    margin        <slot><percent>
    maxreq        <count>
    pmode         [<on/off>]
    power         <slot><on/off>
    primary       <slot>
    ptrace        <slot>
    read          <slot><obj><addr>[<by,wo,lo,bl>]
    reset         <slot>
    secondary     <slot>
    set           <slot><obj1><obj2><value>
    show          <slot><obj1><obj2>
    skippoll      <count>
    skipalt       <count>
    switch        <ena/dis><slot #>
    swreset       <on/off>

Spacebar to continue, 'q' to quit:

    trace         <slot>
    write         <slot><obj><addr><data>[<by,wo,lo,bl>]
    vector        <0,1,2,3,4>
    version       (no args)

    Command+line editing keys:
        ^P: yank previous command
        ^N: yank next command
        ^K: kill to end of line
        ^A: goto start of line
        ^E: goto end of line
        ^B: back one character
        ^F: forward one character
        ^U: kill entire line
        ^D: delete character under cursor
        <ESC>+b: go back one word
        <ESC>+f: go forward one word

    Commands can be abbreviated to any unique command string.

```

H3201

When you enter the **help show** command with no argument, a screen similar to Figure 5-2 is displayed.

Figure 5-2 TCS help show screen

```
TCS HUB<<A>> help show

SYNTAX:
    show <slot><obj1> <obj2>

DESCRIPTION:
    show : cli show command
```

H3202

TCS Hub Tasks

This section describes the commands used to perform tasks from the TCS hub interface.

Connecting to a Card

This procedure describes how to logically connect from a terminal attached to the console/modem I/O ports to a given slot within a LightStream switch. You can connect to a switch card, an NP, or a line card. You can also connect to an NP to access the CLI from the console or modem ports.

Procedure

Step 1 At any TCS prompt, enter:

```
TCS HUB<<A>> connect <slot>
```

where

<slot> = - 10, SA, or SB

Following are some examples of the **connect** command. The first connects to an NP in slot 1:

```
TCS HUB<<A>> connect 1
```

The second connects to a line card in slot 7:

```
TCS HUB<<A>> connect 7
```

Step 2 To exit from connect mode, enter:

```
\.
```

Expected Results

The information displayed when you connect to a card varies depending on the software running in that card at the time the connection is made.

Operational Tips

If you get no response to a **connect** command, you may need to reset the card before you connect to it or download software into it.

Forcing a TCS Hub to Become Primary or Secondary

The TCS hub residing on a switch card controls the switch card itself and acts as a communications hub for the system-wide TCS. In a LightStream switch with two switch cards (SA and SB), the TCS hub on one switch card is the primary TCS hub, and the TCS hub on the other switch card is the secondary TCS hub. The procedure below forces the TCS hub on one switch card to become the primary (or secondary) TCS hub. As you force one TCS hub to become the primary, the other becomes the secondary and vice versa.

Use this procedure before you run diagnostics on a particular switch card. (You can run diagnostics only on the switch card that is *not* primary, as all circuits will be affected.) This procedure does *not* disrupt the flow of traffic through the switch; it only specifies the card that is to be used as the TCS hub.

Note This procedure does not force the switch card to become the primary or secondary switch card. That procedure is described in the *LightStream 2020 Operations Guide*.

Procedure

Step 1 Access the TCS hub interface by connecting a VT100-compatible terminal (or modem) to the TCS console (or modem) port on the back of the LightStream switch. Press the **[Return]** key to obtain the TCS prompt.

Step 2 If you do not see a TCS prompt, enter the following characters:

\ .

Step 3 Determine which TCS hub is the primary or secondary TCS hub by noting the character between the angle brackets (<>) in the TCS prompt. (Refer to Table 5-1 for details.)

Step 4 To force a TCS hub to be the primary TCS hub, enter the following command at the TCS prompt:

```
TCS HUB<<A>> primary <slot #>
```

or, to force a TCS hub to be the secondary TCS hub, enter the following command at the TCS prompt:

```
TCS HUB<<A>> secondary <slot #>
```

where

<slot #> =SA or SB

Expected Results

The TCS hub you specified is set as the primary (or secondary) TCS hub. It takes the TCS approximately 4 seconds to switch the primary and secondary TCS hubs. This process delays only temporarily the servicing of pending TCS slave requests.

Using LightStream Traps

Trap Types and Priorities • Trap Formats • Setting Trap Levels • Viewing Traps • Logging Traps • Customizing the Trap Log and Trap Display • Trap Flowchart

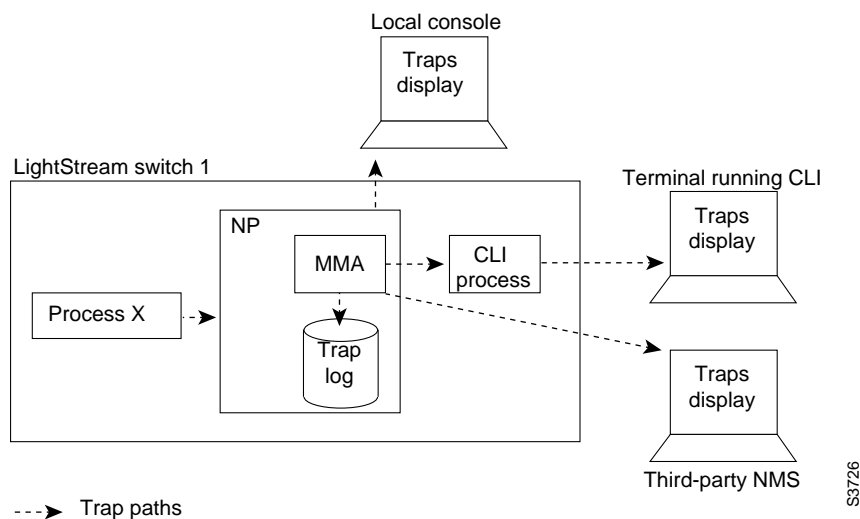
Traps are messages that inform you of network events. A trap may notify you of a serious condition that requires immediate corrective action, or it may give you information that, while important, may not require any action at all. When a network event occurs, the LightStream® switch sends a single, simple trap to the management station. The network operator initiates further interactions with the LightStream switch to determine the nature and extent of the event.

This chapter discusses the traps generated by LightStream switches, their types and formats, categories and priorities, trap logs, trap level settings, and displays. Refer to the *LightStream 2020 Traps Reference Manual* for descriptions of specific traps.

Traps are generated by the processes running on a LightStream switch and are typically displayed on that switch's local console. You can also display traps on another terminal or SPARCstation running the command line interface (CLI) or on a workstation running a third-party network management system (NMS).

Traps move from their initiating process to the master management agent (MMA), the trap log, and the command line interface (CLI) process or third-party network management system (NMS). These paths are shown in Figure 6-1. By default, all traps entering the MMA are logged to a file called `/usr/tmp/mma/mma.traplog`. You can also record the traps for a LightStream switch in a log file on the switch's NP. You can view the file there or you can copy the file to another system and view or process it there. If you use a third-party NMS, you may also be able to record traps in a log file on that system. (Refer to the documentation for your NMS for further information.)

Figure 6-1 Traps are passed from the LightStream processes to the MMA, stored in the trap log, and sent to the CLI process or third-party NMS



By default, all LightStream switches send traps to their local NP (address 127.0.0.1). Each switch has a file (`/usr/app/base/config/mma.trap_communities`) containing that address. To display traps on a different terminal, you can edit the file and *change* the default trap delivery address of the switch to the IP address of the new terminal that is to display the traps. To display traps on an additional terminal, you can *add* the IP address of the additional terminal to the file. For complete instructions on altering the trap delivery address, see the section “Changing the Trap Delivery Address.”

You can also use the CLI to display traps for other switches in the LightStream network. (As noted above, you must change the default trap delivery addresses of those switches to the IP address of the NP that will display their traps.) In this case, two copies of the traps are displayed. The first is displayed on the switch’s local console by an automatic filtering mechanism; the other is displayed on the terminal running the CLI. To prevent the duplicate display, you can disable the filtering mechanism of the local console by setting the MIB object `chassisConsoleTrapLevel` to *off*. You can also change the trap level of this filter to display different levels of traps on the local console. For instructions on setting the filtering mechanism, refer to the section “Setting the Trap Level for the Filtering Mechanism.”

Trap Types and Priorities

LightStream switches generate five categories of traps:

- **SNMP** — This category contains a small number of traps defined by the SNMP MIB-2 specifications. SNMP traps are used by LightStream network operators.

Link Up and Link Down are examples of SNMP traps.

- **Operational (OPER)** — Operational traps provide information on the key system components to help you find and correct problems. Operational traps indicate that something is wrong with the system or that a significant change has occurred in the system status. They can also be used to report the status of the LightStream components. Operational traps are of primary interest to LightStream network operators.

Port 3 down is an example of an operational trap.

- **Informational (INFO)** — These traps provide supplemental details on problems that are reported by some operational and SNMP traps. Informational traps are used by customer support representatives to do advanced troubleshooting and software debugging.

Trunk emtb7.2.5->emtb8.4.2 DOWN [transitioning to down (from has-vci)] is an example of an informational trap.

- **Trace (TRACE)** — These traps are used to track a sequence of actions through a process, allowing an experienced customer support representative to isolate problems. Trace traps are used by customer support representatives to do advanced troubleshooting and software debugging.
- **Debug (DEBUG)** — These traps are used to find and solve serious software problems within a LightStream switch. Debug traps are used only by customer support representatives and developers.

Note Do not turn on debug traps. If you do so, you may affect normal network performance.

SNMP traps have the highest priority, followed by operational, informational, trace, and debug traps, which have the lowest priority (as shown in Figure 6-1).

Table 6-1 Trap Levels

Trap Types (by Priority)	Severity
SNMP	Highest
Operational	t
Informational	t
Trace	t
Debug	Lowest

In most cases, you should display all operational traps and log all operational and informational traps for your network. (These are the defaults).

Trap Formats

This section describes the two trap formats:

- SNMP standard traps
- Enterprise-specific traps

If you are using a third-party NMS to display traps, the display may differ, but the content is the same.

SNMP Standard Traps

The standard SNMP traps include:

- LightStream node name
- System up time when the trap occurred
- Trap name
- Trap generation time
- Port number associated with the trap (if applicable)

Enterprise-specific Traps

Enterprise-specific traps contain the following information:

- LightStream node name

Note The system uses the IP address of the packet containing the trap to look up the name of the node in the /etc/hosts file. If the name is not available, the IP address is displayed.

- System up time when the trap occurred

Note The time is determined by the MIB-2 variable sysUpTime.

- Trap severity level (oper, info, trace, or debug)
- Symbolic trap name

Note This consists of an abbreviation for the software module that generated the trap, followed by a number that uniquely identifies the trap type. (See Table 6-2.) For example, if the symbolic trap name is GIDD_1000, the trap was generated in a process called the Global Information Distribution Daemon (GIDD), and it is an operational trap (1-1999).

Table 6-2 Trap Type Numbering

Trap Type	Numeric Range
Operational	1-1999
Informational	2000-2999
Trace	3000-3999
Debug	4000-4999

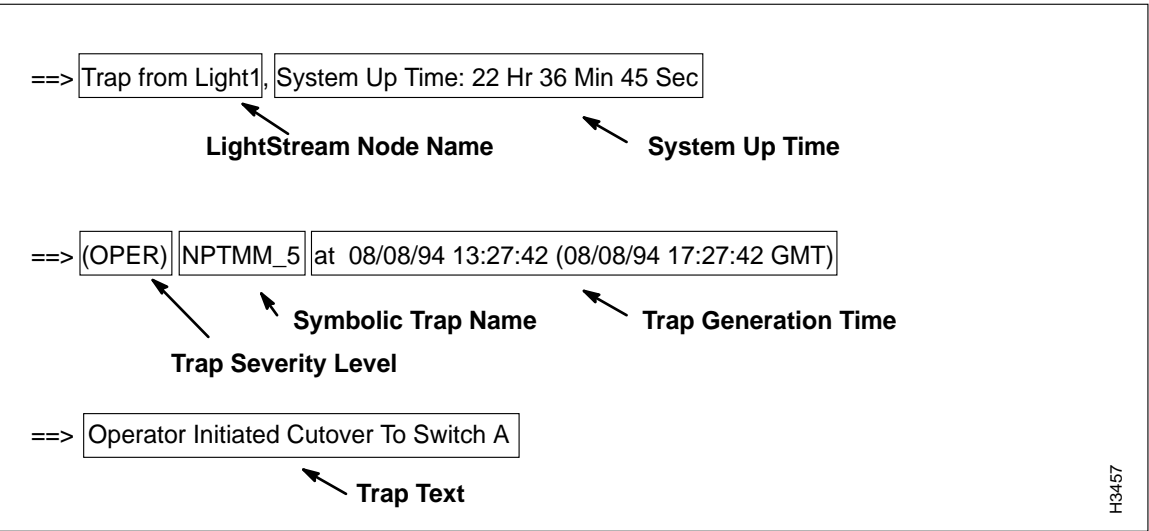
- Trap generation time

Note This is shown in two forms - the time zone you selected during installation and Greenwich Mean Time.

- Trap text

Figure 6-2 shows each field of a sample enterprise-specific trap.

Figure 6-2 The fields included in every LightStream trap



Trace and debug traps also include the process identification (pid) number and process alias name of the process in which the trap occurred.

Setting Trap Levels

As shown in Figure 6-1, traps are generated in processes and passed first to the MMA and then to the CLI (or the third-party NMS) for display. By setting the severity level of the traps that flow between the MMA and the CLI processes, you can specify the traps you want logged in the MMA, displayed on the CLI, or displayed on a third-party NMS. Setting the severity level is referred to as setting the trap level.

You can set the trap levels for:

- All processes running on a LightStream switch — This determines which traps are passed from the processes to the MMA where they are logged in a file.
- Each LightStream MMA (chassis) —This determines which traps are passed by the MMA to the CLI or the third-party NMS.
- The CLI — This determines which traps are actually displayed on the CLI.
- The automatic filtering mechanism — This determines which traps are passed from a local LightStream switch to its console.

You can set the trap level to operational, informational, trace, or debug. You can also turn off all traps at the CLI and on the filtering mechanism. (See Table 6-3.) Although you can change any of these settings as required, the default trap level settings (informational for processes, operational for chassis, and debug for the CLI) are appropriate for most networks.

As shown in Table 6-4, SNMP traps are always displayed since SNMP traps are always passed regardless of the trap level setting. If you set the trap level to operational, operational traps are also displayed. If you set the trap level to a lower priority level, such as trace, the trace traps and all higher priority traps are displayed. When you set the trap level to debug, all levels of traps are displayed.

Table 6-3 Trap Settings, Purposes, Defaults, and Options

Trap		Level		Storage/Display	
Setting	Purpose	Default	Options	Default	Options
Switch Processes	Determines which traps pass from processes to MMA	Info	Info Oper Trace Debug	Log File	

Trap		Level		Storage/Display	
Chassis (MMA)	Determines which traps pass from MMA To CLI	Oper	Info	CLI or NMS	
			Oper		
			Trace		
			Debug		
CLI	Determines which traps display on the CLI	Debug	Info	CLI	
			Oper		
			Trace		
			Debug		
			Off		
Filter Mechanism	Determines which traps pass to local console	None	Info	Local Console	Local Console or CLI Terminal
			Oper		
			Trace		
			Debug		
			Off		

Table 6-4 Trap Level Settings and Displays

Trap Level Setting	Traps Displayed				
	OPER	INFO	TRACE	DEBUG	SNMP
Operational	x				x
Informational	x	x			x
Trace	x	x	x		x
Debug	x	x	x	x	x

Setting the Trap Level for Processes

Setting the trap level for a particular process determines whether the traps generated by that process are passed to the MMA. Traps that are passed from the processes into the MMA are logged in the trap log (if the trap log is enabled).

The default trap level for all processes is informational. This level is appropriate for most applications.

Follow the procedure below to set the trap level for processes.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the cli> prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 3 If you do not know which processes are running on this switch, enter the following at the cli> prompt to display a list of the processes:

```
cli> walksnmp lwmaTrapCliAlias
```

This command lists the pid numbers and alias names of all the processes running on this LightStream switch. The pid numbers follow the term “Name: lwmaTrapCliAlias.” and the alias names follow the term Value, as the following screen output shows:

```
cli> walksnmp lwmaTrapCliAlias
Name: lwmaTrapCliAlias.3 Value: CAC
Name: lwmaTrapCliAlias.4 Value: GIDD
Name: lwmaTrapCliAlias.5 Value: NPCC
Name: lwmaTrapCliAlias.6 Value: LCC3
Name: lwmaTrapCliAlias.7 Value: LCC9
Name: lwmaTrapCliAlias.8 Value: LCC5
Name: lwmaTrapCliAlias.10 Value: LCC7
Name: lwmaTrapCliAlias.37 Value: ND
Name: lwmaTrapCliAlias.40 Value: TRAPMON
Name: lwmaTrapCliAlias.43 Value: lcmon (secondary
Name: lwmaTrapCliAlias.44 Value: trunkmon
Name: lwmaTrapCliAlias.45 Value: cardmon
Name: lwmaTrapCliAlias.46 Value: RMON
Name: lwmaTrapCliAlias.47 Value: KLOG
Name: lwmaTrapCliAlias.48 Value: NPTMM
Name: lwmaTrapCliAlias.49 Value: COLLECTOR. . .
```

Select the process(es) you want from the list. The *LightStream 2020 System Overview* describes each of these processes.

Step 4 To set the trap level for a selected process, enter the following at the cli> prompt:

```
cli> set pid {<#>|<alias>} traplevel <value>
```

where

- {<#>|<alias>} = The process number or alias name.
- <value> = oper, stepinfo (default), trace, or debug

Step 5 Verify the process trap level by entering the following at the cli> prompt:

```
show {pid <#>|<alias>} traplevel
```

Expected Results

The trap level for the specified process is set to the appropriate level.

Setting the Trap Level for the MMA

The trap level for the MMA determines which traps are passed to the CLI and third-party NMS. The traps that are passed to the CLI process from the MMA must pass the CLI trap level setting to be displayed, but those passed to the NMS from the MMA are displayed on the NMS, unless the NMS has its own filtering capabilities.

The default trap level for the chassis is operational. This level is appropriate for most networks.

The trap level for the chassis is normally set during network configuration. If you want to temporarily change the configured setting, use the procedure below. (The change will be lost if the switch reboots.) If you want to make a permanent change to the MMA Trap Filter Level, use the LightStream configurator to edit the configuration and download the changes to the appropriate devices as described in the *LightStream 2020 Configuration Guide*.

Procedure

- Step 1** Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

- Step 2** Set the SNMP community to a read/write community by entering the following at the cli> prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

- Step 3** Set the trap level for a chassis by entering the following at the cli> prompt:

```
cli> set chassis traplevel <value>
```

where

<value> =oper (default), info, trace, or debug

- Step 4** Verify that the trap level has been changed by entering the following at the cli> prompt:

```
cli> show chassis agent
```

Review the value of the MMA Trap Filter Level attribute.

Expected Results

The trap level for the specified chassis is changed.

Setting the Trap Level for a CLI Session

This section tells you how to set the trap level for a CLI session. The trap level setting for the CLI determines which traps are taken from all the different LightStream switches (chassis) in the network and displayed by the CLI.

The default trap level for the CLI is debug. This means that all traps (of all levels) that can pass from the processes to the MMA and into the CLI are displayed. This level is appropriate for most applications.

Procedure

- Step 1** Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 2 Set the trap level for the CLI by entering the following at the cli> prompt:

```
cli> set cli traplevel <value>
```

where

<value> =off (displays no traps), oper, info, trace, or debug (default)

Step 3 Verify that the trap level for the CLI has been changed by entering the following at the cli> prompt:

```
cli> show cli traplevel
```

Expected Results

The CLI trap level is changed to the level you specified.

Setting the Trap Level for the Filtering Mechanism

The trap level setting for the filtering mechanism in each LightStream switch determines which traps are displayed on the local console (if the switch has a local console). The filtering mechanism begins automatically whenever a switch is started.

The default trap level for the filtering mechanism is info. This means that all informational and operational traps generated on the local switch are displayed on the local console. This level is appropriate for most applications.

Follow the procedure below to set the trap level for the filtering mechanism.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the cli> prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 3 Set the trap level for the CLI by entering the following at the cli> prompt:

```
cli> set chassis consoletraplevel <value>
```

where

<value> =off (displays no traps), oper, info (default), trace, or debug

Step 4 Verify that the CLI trap level has been changed by entering the following at the cli> prompt:

```
cli> show chassis general
```

Expected Results

The CLI trap level is changed to the level you specified.

Viewing Traps

You can view traps from the CLI or from a third-party NMS. Using the CLI, traps are displayed in the format described in the section “Trap Formats.” Trap messages may be interleaved with the other information displayed by the CLI, depending on how your system is set up. If you use a third-party NMS, the message format may vary from the format described earlier, but the content is the same. (Refer to the third-party NMS documentation for information on viewing traps.)

The traps that appear on the CLI are determined by the CLI trap level that has been set and the trap delivery addresses that are defined for each switch. The MMA trap level determines which traps are passed to the third-party NMS for display.

Figure 6-3 shows a typical CLI display including traps.

Figure 6-3 Typical CLI traps display

```
cli> show card 1 all
Card Name: NP1
Card PID: 3
Operational Status: Up
Administrative Status: Up
Configuration Register: Up

LC Software Version:
LCC Software Version: npcc: compiled 1 Oct 1994 @4:40:22

Card Type: NP
Top Temperature: 84 F(28 C)
Bottom Temperature: 105 F (40 C)
TCS Voltage: 4.980 volts
VCC Voltage: 4.980 volts
SCSI Voltage: 4.687 volts
This card has no access or trunk ports
cli>

==>Trap from emtblnpl.lscf.com, System up time: 20 Hr 47 Min 11 Sec
==> (OPER) NPTMM_6 at 08/16/94 07:44:35 EDT (08/16/94 11:44:35 GMT)
==> TEMPERATURE#2 (105.468F) of card 1 is outside the normal range

cli> set card 1 active
cli> show card 2 all
Card Name: LowSpeedEdge
Card PID: 12
Operational Status: Up
. . .
cli>
```

S3719

Logging Traps

You can log the traps that occur on each LightStream switch. The traps are logged on the NP of the switch in a file called `mma.traplog` in the `/usr/tmp/mma` directory. This circular file can store approximately 6000 traps before the oldest trap is overwritten by the newest trap. The priority of the logged traps is determined by the trap levels set for each process. This section tells you how to enable, disable, and view the trap log.

You may also be able to log traps from your third-party NMS. Refer to the third-party NMS documentation for more information.

Enabling or Disabling the Trap Log

This section tells you how to enable or disable the trap log for a particular LightStream switch. Traps cannot be logged unless the trap log is enabled.

The default setting for the trap log is enabled (on). This setting is appropriate for most networks.

You usually specify whether the trap log is enabled or disabled during network configuration. If you want to temporarily change the configured setting, use the procedure below. If you want to make a permanent change to the MMA Trap Log Status, use the LightStream configurator to edit the configuration and download the changes to the appropriate devices as described in the *LightStream 2020 Configuration Guide*.

Note If you disable the trap log for a particular switch, the traps for that switch are not recorded in a file. If a problem occurs on that switch, you will not have a record of the traps that were reported.

Procedure

Step 1 Verify that the target switch is correct by entering the following at the `cli>` prompt:

```
cli> show chassis general
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the `cli>` prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 3 Enable or disable the trap log for a particular chassis by entering the following at the `cli>` prompt:

```
cli> set chassis traplog <value>
```

where

<value> =on (enables the trap log [default]) or off (disables the trap log)

Step 4 Verify that the traplog has been enabled or disabled for a particular chassis by entering the following at the `cli>` prompt:

```
cli> show chassis agent
```

Review the value of the MMA Trap Log Status attribute.

Expected Results

The trap log for the specific chassis is enabled or disabled, as specified.

Viewing the Trap Log

If you have logged traps in the NP of a particular LightStream switch, those traps are recorded in a circular file called `/usr/tmp/mma/mma.traplog`. You can view the trap log by accessing that file from the CLI or the LynxOS shell, or you can copy the file to a third-party NMS or a workstation and view it there. This section tells you how to view the trap log. It also describes how to copy the trap log so that it can be viewed from another system.

Procedure 1: Viewing Traps from the bash Prompt

Display the trap log file from the LynxOS shell by entering the following at the `bash#` or `bash$` prompt:

```
bash$ cbufpr [-hv] [-all] [-tail] -<number> [-f] [-trap] traplog |more
```

where

- `[h]` = Displays this help message.
- `[v]` = Displays `cbufpr` version information.
- `[all]` = Allows you to read files of all formats, including files that are not circular.
- `[tail]` = An optional argument that displays the last 20 lines of the traplog file (the lines containing the most recent traps). If you do not enter this argument, the entire traplog file is displayed. If the file is long, it scrolls across the screen until reaches the end.
- `[f]` = Continue reading from end of file rather than exiting. The switch allows you to display traps as they accumulate while you are viewing other parts of the file.
- `[number]` = Specifies the number of lines to display. This switch can be used with the `-tail` switch to specify the number of lines displayed from the bottom of the file.
- `|more` = Display one page at a time. Press the spacebar key to display the next page. If you do not use `|more`, the file will scroll across the screen.

Procedure 2: Viewing Traps from CLI Protected Mode

Step 1 At the `cli>` prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 To display the log file, enter the following at the `*cli>` prompt:

```
*cli> shell "cbufpr [-hv] [-all] [-tail] -<number> [-f] [-level] <traplog> |more"
```

For further information on the `cbufpr` command, refer to the *LightStream 2020 Operations Guide*.

Procedure 3: Moving the Trap Log from the NP

Step 1 Before attempting to move the trap log from the NP, obtain a user name and password for an account on the workstation or host where you want to place the trap log.

Step 2 At the cli> prompt, enter:

```
cli> protected
```

Step 3 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 4 At the *cli> prompt, enter:

```
*cli> shell "ftp <IP address of the workstation or host where you want to place the trap log>"
```

You are prompted to log in to the workstation or host.

Step 5 Log in to the workstation or host.

Step 6 To place the trap log file in any directory other than the login directory on the workstation or host, enter **cd <directory name>** to change to the correct working directory.

Step 7 Enter the following at the ftp> prompt:

```
ftp> put /usr/tmp/mma/mma.traplog [<new name>]
```

where

[<new name>] = The file name identifying the chassis or the appropriate directory name for the file. For example, if you are moving a trap log for a switch called Light5, the new name could be mma_Light5.traplog.

This command sends the log file to the specified workstation or host. The system tells you when the file transfer is complete.

Step 8 Enter the following at the ftp> prompt:

```
ftp> quit
```

Step 9 Use any **more** or **cat** command or a screen editor such as emacs or vi to view the mma.traplog file on the workstation or host.

Expected Results

Following is an example of a LightStream switch trap log:

Note The LightStream switch name does not appear in the trap log because all traps in a trap log apply to a single LightStream switch.

```
(INFO) NDD_2004 at 08/26/94 14:02:35 EDT (08/02:35 GMT)
Trunk emtb7.5.0->emtb5.5.0 UP [transitioning to has-vci (from managed)]
(OPER) NPTMM_6 at 08/26/94 14:02:51 EDT (08/26/94 18:02:51 GMT)
TEMPERATURE#2 (103.515F) of card 1 is outside the normal range
(INFO) NDD_2005 at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Trunk emtb7.5.1->emtb5.5.1 DOWN [transitioning to DOWN (from has-vci)]
Link Down Trap at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Port 5001
Link Down Trap at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
```

```
Port 3006
Link Down Trap at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Port 3005
(OPER) CAC_1007 at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Switch Device Write Error 28
Link Down Trap at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Port 4001
Link Down Trap at 08/26/94 14:03:40 EDT (08/26/94 18:03:40 GMT)
Port 4000
. . .
```

Customizing the Trap Log and Trap Display

In addition to setting the trap levels to determine which traps are logged and displayed, you can customize your trap log and trap display by turning specific traps on or off. You can perform the following functions:

- Display the status of every trap for a particular process or a LightStream switch to track any changes you have made
- Turn a trap on or off in a particular process
- Turn a trap on or off in all processes (turn a trap on or off globally)
- Enable or disable a trap in a particular node (globally)

Note The procedures described here are used for advanced troubleshooting and software debugging. The commands to turn particular traps on and off are available from protected mode only and should be used by experienced users only.

The **show trap** command displays the status of each trap for a particular process or switch so that you can keep track of any changes you make. The procedure to display the status of each trap is also provided in this chapter.

Turning traps on in one or all processes with the **set trap** command allows selected traps to be passed to the MMA, even if their severity level is lower than the trap level set for the process in which they were generated. This allows you to override the process trap level and pass specific traps from specific processes to the MMA where they are logged. Disabling traps is used to turn off a particular trap if you no longer need notification.

Figure 6-5 shows the flow of traps through the switch. You may want to refer to the figure while reading the procedures.

Displaying the Status of All Traps for a Process or Chassis

This section tells you how to view the status of every trap within a particular process or for an MMA. You can view this status to keep track of any customizations that may have been made to the traps.

Procedure 1: View Status of One or More Traps for a Process

Step 1 Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the cli> prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 3 At the cli> prompt, enter:

```
cli> show trap pid {<#>|<alias>} <trap #> [<group name>]
```

where

- {<#>|<alias>} = The number or the alias name of the process. Use **walksnmp lwmaTrapCliAlias** to obtain the pid numbers and the pid aliases for all processes running on the LightStream switch.
- <trap #> = The number of the trap(s). Enter the symbolic trap name or trap number. You can set multiple traps by specifying the trap numbers (ndd_3 ndd_14 lcc_5), or by using the wild card character "*" to specify all traps for a particular process ("ndd*" for all NDD traps).
- [<group name>] = An optional argument used to define a group of traps on which you want to show status. To use this argument, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. For instructions on creating the cli.groups file, refer to the section “Creating the cli.groups File.”

Procedure 2: View Status of One or More Traps for an MMA

Step 1 Verify that the target switch is correct by entering the following at the cli> prompt:

```
cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 2 Set the SNMP community to a read/write community by entering the following at the cli> prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 3 At the cli> prompt, enter:

```
cli> show trap <trap #> [<group name>]
```

where

- <trap #> = The number of the trap(s). Enter the symbolic trap name or trap number. You can set multiple trap numbers by specifying the trap numbers (ndd_3 ndd_14 lcc_5), or by using the wild card character "*" to specify all traps for a particular process ("ndd*" for all NDD traps).

- [<group name>] = An optional argument that defines a group of traps. To use this argument, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. For instructions on creating the cli.groups file, refer to the section “Creating the cli.groups File.”

Expected Results

If you enter **show trap ndd_3 ndd_4 ndd_5 ndd_1001**, the status of these three traps is displayed as follows:

Figure 6-4 Sample specific trap status display

```
*cli> show trap ndd_3 ndd_4 ndd_5 ndd_1001
Trap NDD_3: off - enabled
Trap NDD_4: off - enabled
Trap NDD_5: off - enabled
Trap NDD_1001: off - enabled
*cli>
```

H3471

If you enter **show trap "*"**, the status of all traps in the MMA is displayed as follows:

```
cli> show trap "*"
Trap GENERIC_TEST (1): off
Trap TRUNKMON_1 (2): off
Trap NDD_1 (3): off
Trap NDD_2 (4): off
Trap NDD_3 (5): off
Trap NDD_4 (6): off
Trap NDD_5 (7): off
Trap NDD_6 (8): off
Trap NDD_7 (9): off
Trap NDD_8 (10): off
Trap NDD_1000 (11): off
Trap NDD_1001 (12): off
Trap NDD_1002 (13): off
Trap NDD_2000 (14): off
Trap NDD_2001 (15): off

. . .
```

The example above shows a partial trap display. Several screens of traps are actually displayed when you issue this command.

Turning a Trap On or Off in a Specific Process

This section tells you how to turn a trap on or off in a particular process running on a LightStream switch. In some cases you may have multiple instances of a process running. This procedure allows you to see the traps from a specific process.

The default for all traps in all processes is off. Turning a trap on in a particular process allows it to be passed to the MMA where it will be logged, even if it has a severity level below the trap level set for the process. This allows you to display a particular trap without displaying or logging all traps at that severity level. This procedure is especially important during troubleshooting and debugging.

For example, if the trap level for process number 17 is *info* and you turn on a trap with a severity level of trace in process number 17, that trace trap is passed from the process to the MMA whenever it occurs, even though its severity level is lower than the trap level for the process. Setting a trap to *on* in a process overrides the trap level setting and allows the trap to be passed. However, if the trap is set to *off* (the default), the trap is passed to the MMA only if its severity level is equal to or higher than the trap level setting of the process. (The trap does not override the trap level setting.)

Before a trap can pass from the process to the MMA, the LightStream switch checks the process state. The trap is passed to the MMA and logged if one of the following conditions exists:

- The trap has been turned on in this process.
- The trap has been turned on globally (see the next procedure).
- The trap severity level is equal to or greater than the process trap level.

Procedure

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Verify that the target switch is correct by entering the following at the cli> prompt:

```
*cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 4 Set the SNMP community to a read/write community by entering the following at the *cli> prompt:

```
*cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 5 To determine which processes are running on this node, enter the following at the *cli> prompt:

```
*cli> walksnmp lwmaTrapCliAlias
```

This command lists the pid numbers and alias names of all the processes running on this LightStream switch. The pid numbers follow the term Name: lwmaTrapCliAlias and the alias names follow the term Value.

Find the process you want in this list. The *LightStream 2020 System Overview* describes each of these processes.

Step 6 At the *cli> prompt, enter:

```
*cli> set trap pid{<#|alias>} {on|off} <trap #> [<group name>]
```

where

- {<#>|<alias>} = The process number or alias name. Use **walksnmp lwmaTrapCliAlias** to obtain the pid numbers and the pid aliases for all processes running on the LightStream switch.

- {on|off} Specifies whether the trap is on or off. The default is off.
- <trap #> = The trap(s) you want to turn on or off. Enter the symbolic trap name or trap number. You can set multiple trap numbers by specifying the trap numbers (ndd_3 ndd_14 lcc_5), or by using the wild card character "*" to specify all traps for a particular process ("ndd*" for all NDD traps).
- [<group name>] = An optional argument that defines a group of traps. To use this argument, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. For instructions on creating the cli.groups file, refer to the section "Creating the cli.groups File."

Step 7 To display the status of each trap in the selected process, enter the following at the *cli> prompt:

```
*cli> show trap pid {<#>|<alias>} ""
```

Expected Results

The trap(s) that you have turned on pass to the MMA whenever that trap is generated in the selected process. Traps that are turned off pass to the MMA only if they have an equal or higher severity level than the process trap level.

Turning a Trap On or Off in All Processes (Global)

This section tells you how to turn a trap on or off in all processes in a particular LightStream switch. This is referred to as turning traps on or off *globally*. Turning a trap on globally allows it to be passed from any process to the MMA, even if it has a severity level below the trap level set for the processes. The default for all traps in all processes is off.

This procedure allows you to display particular traps without having to display or log all traps at that severity level. This procedure is especially important during troubleshooting and debugging.

For example, if the trap level for the ndd process is info and you turn on a trap with a severity level of trace in the ndd process, that trace trap is passed from the process to the MMA whenever it occurs, even though its severity level is lower than the trap level for the process. Setting a trap to *on* in all processes overrides the trap level setting and allows the trap to be passed. However, if a trap is set to *off* in a process (the default), the trap is passed to the MMA only if its severity level is equal to or higher than the trap level setting of the process.

Before a trap can pass from a process to the MMA, the LightStream switch checks the process state. The trap is passed to the MMA and logged if one of the following conditions exists:

- The trap has been turned on in this process (see the previous procedure).
- The trap has been turned on globally.
- The trap severity level is equal to or greater than the process trap level.

Procedure

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Verify that the target switch is correct by entering the following at the cli> prompt:

```
*cli> show chassis
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 4 Set the SNMP community to a read/write community by entering the following at the cli> prompt:

```
cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 5 At the *cli> prompt, enter:

```
*cli> set trap global {on|off} <trap #> [<group name>]
```

where

- {on|off} Specifies whether the trap is on or off. The default is off.
- <trap #> = The trap(s) you want to turn on or off. Enter the symbolic trap name or trap number. You can set multiple trap numbers by specifying the trap numbers (ndd_3 ndd_14 lcc_5), or by using the wild card character "*" to specify all traps for a particular process ("ndd*" for all NDD traps).
- [<group name>] = An optional argument used to define a group of traps that you want to turn on or off. To use this argument, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. For instructions on creating the cli.groups file, refer to the section “Creating the cli.groups File.”

Step 6 Display the status of each trap in all processes by entering the following at the cli> prompt:

```
cli> show trap ""
```

Expected Results

The trap(s) that you have turned on are passed to the MMA whenever that trap is generated in any process. Traps that are turned off are passed to the MMA only if they have an equal or higher severity level than the trap level for all processes.

Enabling/Disabling Traps for a Specific LightStream Switch

This section tells you how to enable or disable specific traps for a particular LightStream switch. This procedure allows you to disable a particular trap in a switch to prevent it from being displayed. You may choose to do this if a particular trap recurs regularly and you feel that its display is unnecessary.

In this situation, the LightStream switch does not check to see if a trap is enabled or disabled until it reaches the MMA from a process. The MMA checks to see if the trap is enabled or disabled. If the trap is disabled, it is discarded. If the trap is enabled and if its severity level is equal to or greater than the MMA trap level, the trap is passed to the CLI or the third-party NMS.

Procedure

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Verify that the target switch is correct by entering the following at the *cli> prompt:

```
*cli> show snmp
```

If you need instructions on changing the target switch, refer to the “Getting Started” chapter of the *LightStream 2020 Operations Guide*.

Step 4 Set the SNMP community to a read/write community by entering the following at the *cli> prompt:

```
*cli> set snmp community <community name>
```

where

<community name> = The SNMP read/write community that you want to access.

Step 5 At the *cli> prompt, enter:

```
*cli> set trap {enable|disable} <trap #> [<group name>]
```

where

- {enable|disable} Specifies whether the trap is enabled or disabled. The default is enabled.
- <trap #> = The trap(s) you want to turn on or off. Enter the symbolic trap name or trap number. You can set multiple trap numbers by specifying the trap numbers (ndd_3 ndd_14 lcc_5), or by using the wild card character "*" to specify all traps for a particular process ("ndd*" for all NDD traps).
- [<group name>] = An optional argument used to define a group of traps that you want to turn on or off. To use this argument, the group must be defined in an ASCII file called cli.groups in the /usr/app/base/config directory. For instructions on creating the cli.groups file, refer to the section “Creating the cli.groups File.”

Step 6 Display the status of each trap in the MMA by entering the following at the *cli> prompt:

```
*cli> show trap ""
```

Expected Results

The disabled trap for the selected node is never passed to the CLI or displayed on a third-party NMS. The display shows traps are either on, off, or disabled. (If the status is either on or off, that implies the trap is enabled. Otherwise the trap is disabled.)

Trap Flowchart

This flowchart shows how traps can be customized and when they are either passed forward or discarded. The text to the right shows the CLI commands that you can use to customize the display and trap log.

Figure 6-5 How traps are passed through the system

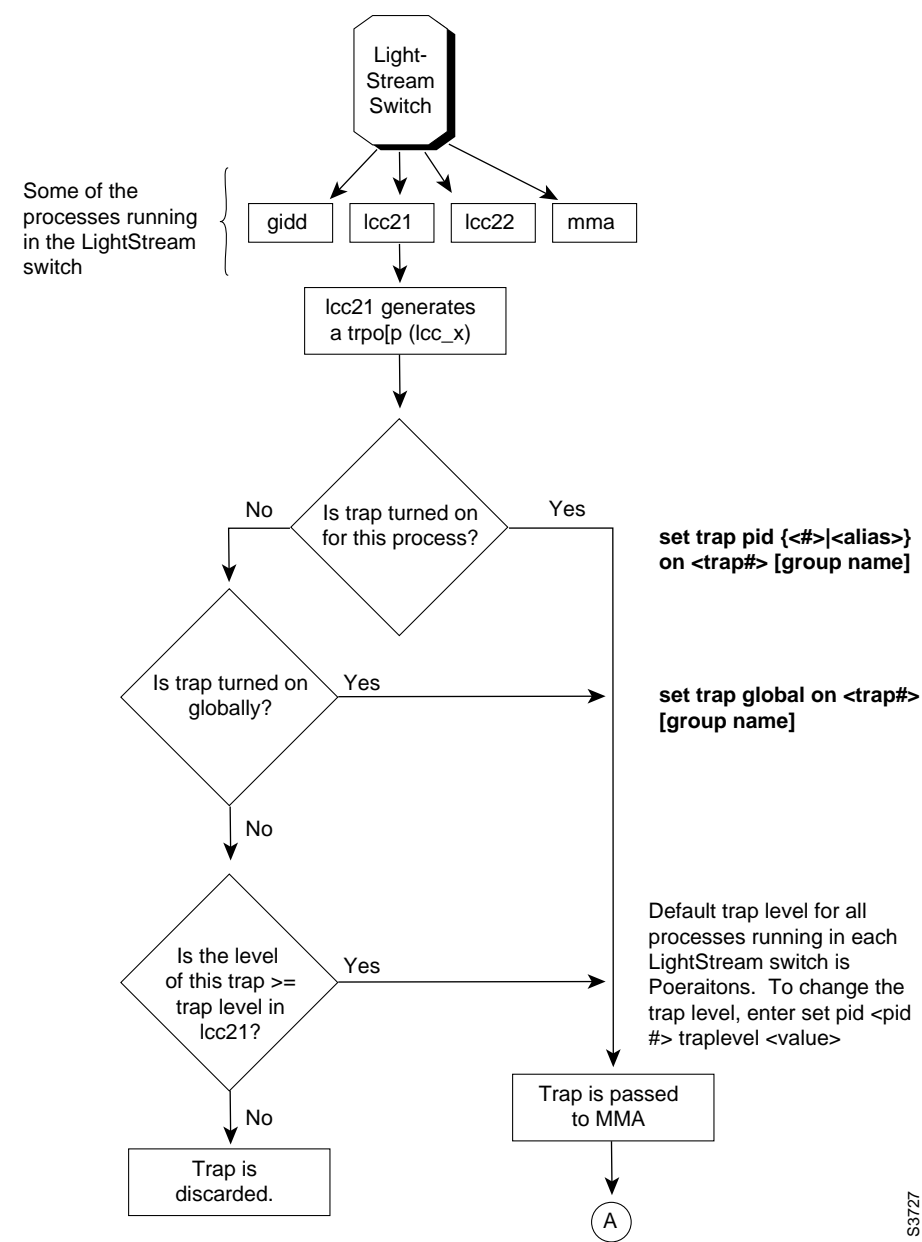
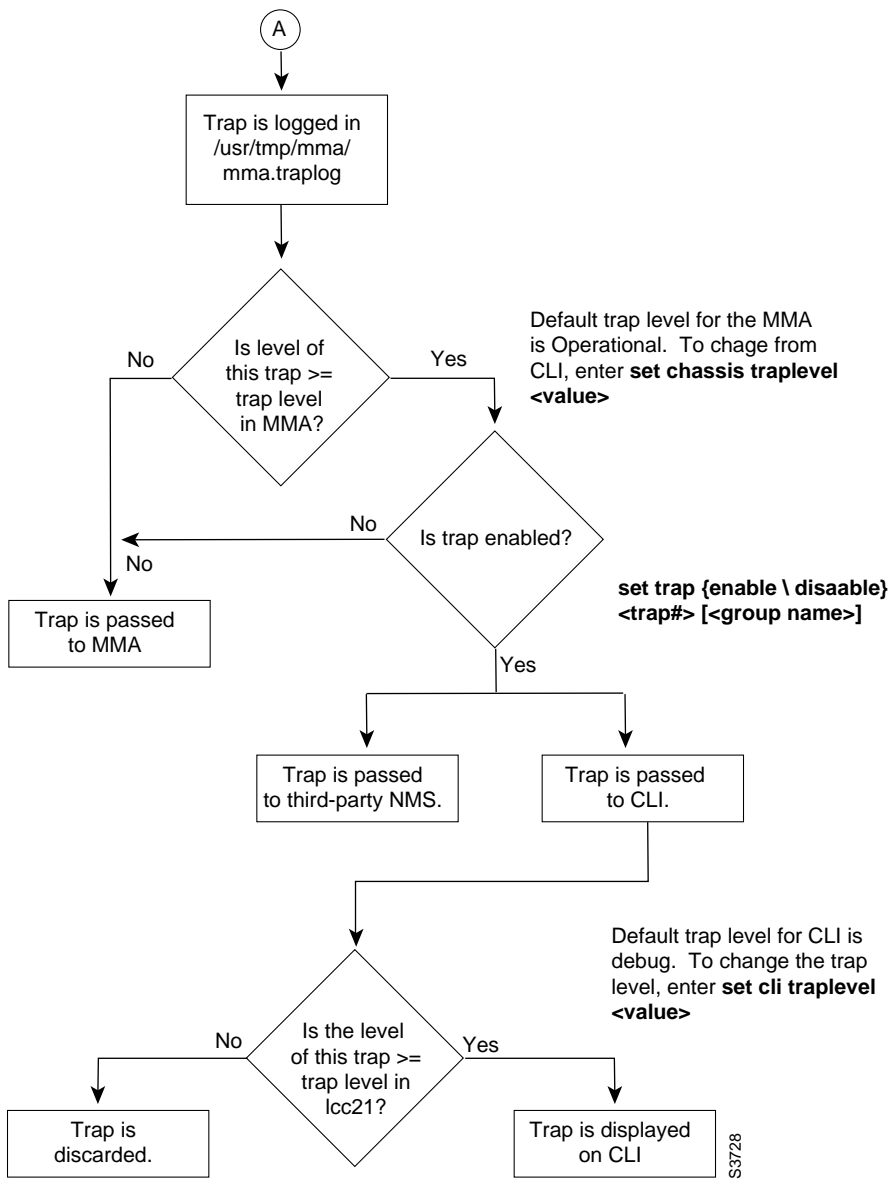


Figure 6-6 How traps are passed through the system (Concluded)



Troubleshooting

Trunk Line Monitoring • Isolating Trunk Problems • Diagnosing Port Problems

This chapter describes trunk line monitoring and some troubleshooting procedures that you can follow if you encounter a problem.

Trunk Line Monitoring

Loss of data can cause external protocols to retransmit, resulting in network congestion and delays. To avoid loss of data, line quality must be kept very high. The line card control process (LCC) continually monitors line quality on each trunk using the switch's Trunk Up-Down (TUD) protocol. This protocol detects a trunk that is down or line quality that is poor. When a trunk that was down comes back up, the TUD protocol returns it to service.

The TUD protocol monitors line quality by having each switch send short test messages at regular intervals. When a switch starts up, it begins sending TUD messages down each trunk that it supports. If the local switch receives TUD messages consistently from the remote switch, it declares the trunk up.

If a switch misses an established number of TUD messages from a trunk, it declares the trunk down. When the trunk is declared down, a trap is displayed indicating the change of status. The following is a typical trunk down message:

```
Link down trap from Light7, System Up Time: 23 Hr 29 Min 50 Sec Port: 5.0
```

When the trunk comes up again, the switch sends another trap. The following is a typical trunk up message:

```
Link up trap from Light7, System Up Time: 23 Hr 29 Min 55 Sec Port: 5.0
```

Isolating Trunk Problems

A trunk may go down temporarily and come back up shortly without intervention. However, if the trunk remains down or transitions constantly in and out of service, you must find and correct the problem. Use the *loopback tests* described in this section to isolate the faulty component. A loopback test is a software or hardware test that alters the flow of data so that an electronic signal is returned to its sender.

Trunks can be directly connected or they can be connected with data service unit/channel service unit (DSU/CSU) devices through telephone lines. Figure 7-1, Figure 7-2, and Figure 7-3 illustrate the major components of the switch-to-switch connection over a trunk. A trunk interface loop occurs in the circuitry within the switch and does not involve components external to the switch.

Figure 7-1 Components of a telco connection between low-speed line cards in two switches

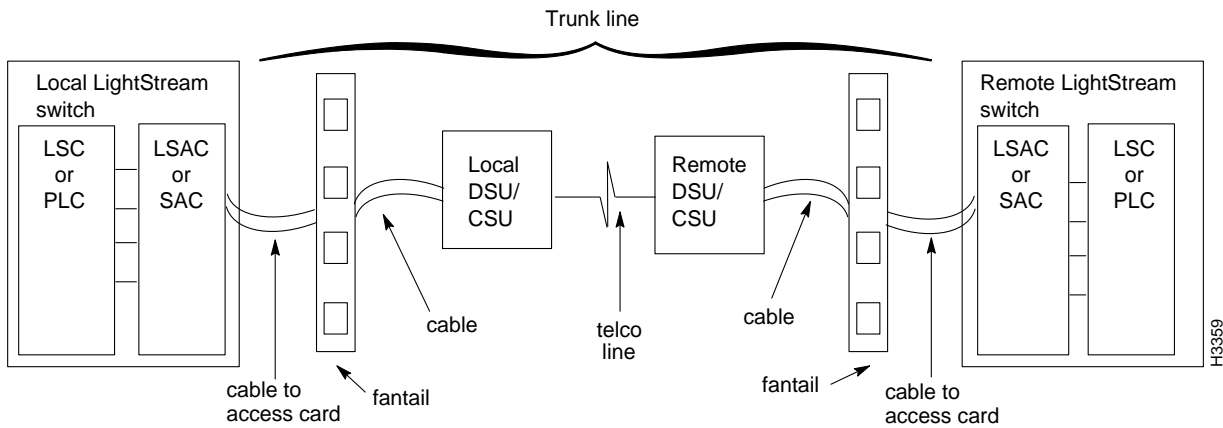


Figure 7-2 Components of a direct connection between low-speed line cards in two switches

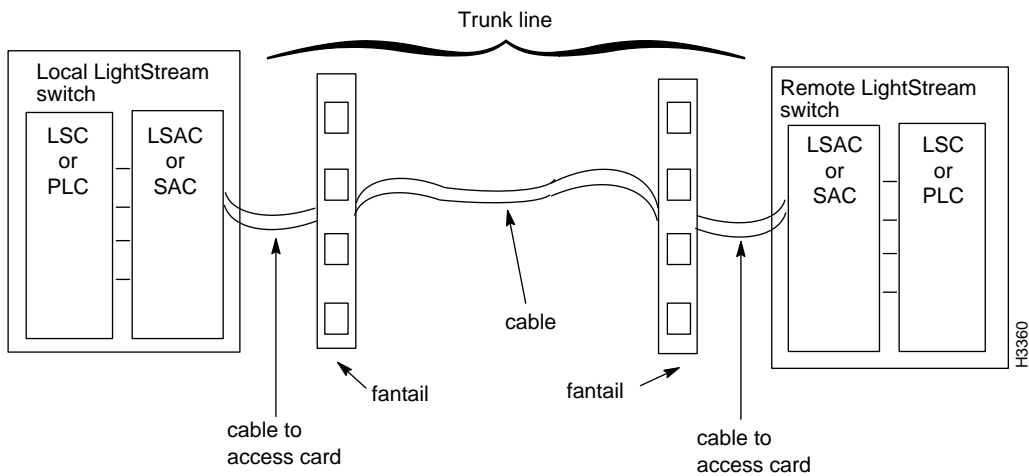
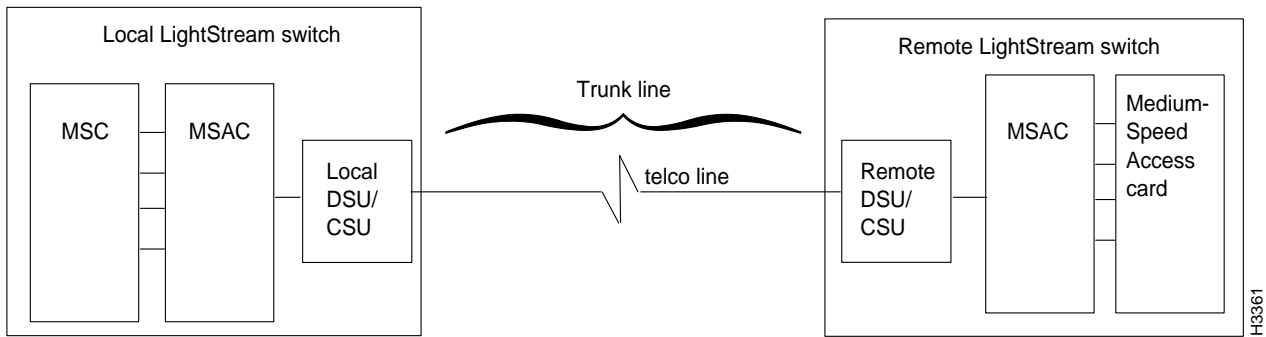


Figure 7-3 Components of a connection between medium-speed line cards in switches with integral DSU/CSUs



Fault Isolation for Link Down Events

Most trunk failures are temporary, caused by problems on the telephone company line. A trunk is usually returned to service within 3 minutes without any intervention on your part and before the telephone company finds anything wrong. If a trunk is reported down, you should wait at least 10 minutes to make sure that the problem is not temporary.

If the trunk does not come up within 10 minutes, you must identify the failed portion of the trunk. To do this, you run a series of loopback tests to segment the trunk from end to end, starting at the I/O board of the switch and progressing out from the switch. This progressive process tests each segment in sequence to find the exact location of the failure.

Looping Ports

The loopback tests allow you to pinpoint a fault by successively looping a signal at various points. The LightStream switch provides three loopback tests: external, remote, and internal.

You can loop any port; however, only frame relay ports and trunk ports have active port management protocols that automatically verify the port's ability to process data. The first procedure in this section tells you how to run a loopback test on trunk ports (ports that connect LightStream systems). The second procedure tells you how to run a loopback test on edge ports.

External Loopback Test

The external loopback test loops data from the line card to the line chip or to the Physical Layer Protocol Processor (PLPP) I/O module to see if the relevant I/O module is working correctly. This test determines if the I/O module is able to successfully encode and decode the data that it receives from the line card.

Remote Loopback Test

The remote loopback test loops data from an external device through the I/O module and back. This test verifies that the data sent from the remote end can cross the telco line or cable, pass through the switch, and return to the remote end.

Internal Loopback Test

The internal loopback test loops data from the line card to the line chip or to the PLPP I/O module to see if the relevant I/O module is able to receive intact data. If this loop is successful, you know that the data can reach the I/O module properly. However, you do not know if the I/O module correctly encodes the data that will be sent out onto the line.

The line chip and the PLPP I/O module take bytes from the line card and convert them into the required form to be sent out from the access card. The line chip encodes frames or cells into SDLC frames. The PLPP I/O module encodes cells into the ATM standard cell payload scrambling format for T3 or E3, depending on the access card. (Cell payload scrambling is sometimes referred to as PLCP scrambling.)

Note Always call the sites involved and ask permission before looping a trunk.

Note Use the trap log and an up-to-date network diagram to determine the success or failure of your loops. The network diagram helps you to visualize the network and makes troubleshooting easier.

Procedure 1: Looping Trunk Ports

This procedure tells you how to loop data through a trunk between two LightStream trunk ports. If you receive an indication that data is not passing on a trunk between two trunk ports, follow this procedure to run an external loop on one of the trunk ports.

Step 1 Type the following at the cli> prompt:

```
cli> set port <port#> loop external
```

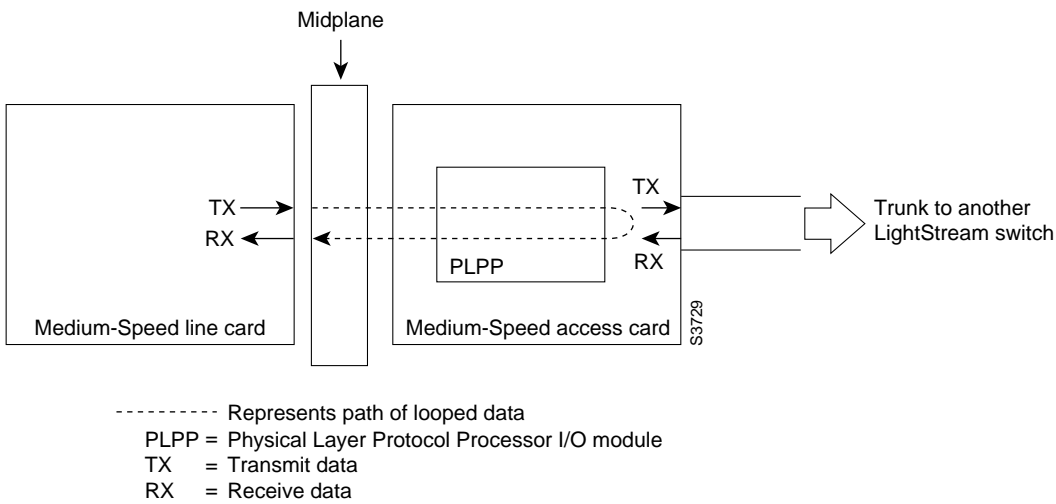
where

<port#> = The port you want to loop. The port number is in card.port format (card = 2 - 10; port = 0 - 7).

The command line interface (CLI) automatically sets the administrative status of the selected port to *testing* and starts the loopback test. Figure 7-4 shows how the data is looped during an external loopback test.

If the external loop succeeds, the trunk comes up. A trap stating that the trunk is up is reported and the Operational Status of the port changes to *up*. This means that the local trunk is configured correctly and that you have come up in a testing situation. Proceed to Step 2.

Figure 7-4 An External Loop on a Trunk Port



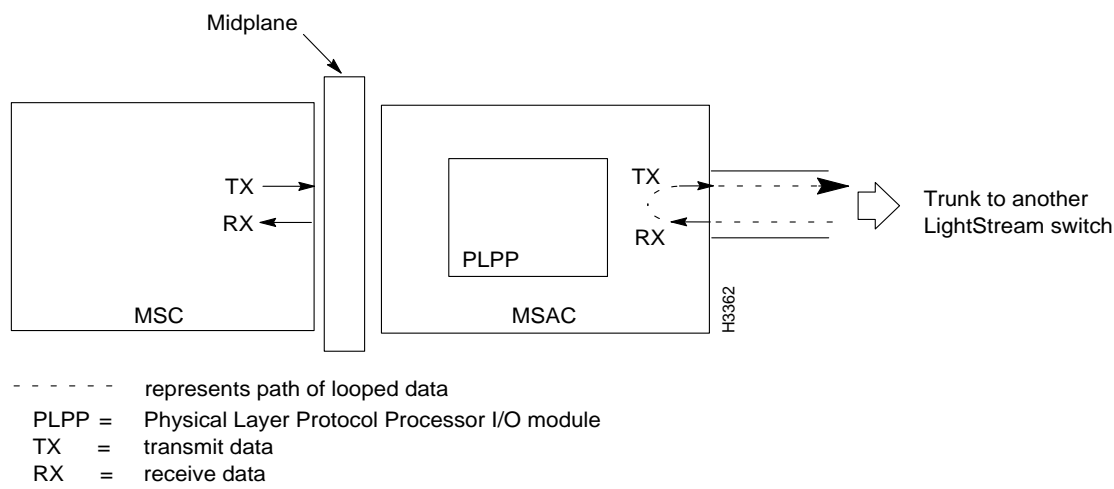
If the external loop fails (the trunk does not come up), the trunk is not configured properly or a hardware problem may exist. Proceed to Step 3.

Step 2 Run a remote loop on the port by typing the following at the cli> prompt:

```
cli> set port <port#> loop remote
```

The CLI automatically sets the Administrative Status of the selected port to *testing* and starts the loopback test. Figure 7-5 shows how the data is looped during a remote loopback test.

Figure 7-5 A remote loop on an MS trunk port



Note The loop remote option sets the local port into a remote loop for the remote port. It does not loop the remote port. Verification of a successful remote loopback must be made at the remote system.

If the remote loop succeeds, the trunk port reports up at the remote end. In this case, it is likely that a problem with the local port which is providing the remote loop exists. While providing the remote loopback, the local port displays an Operational Status of down.

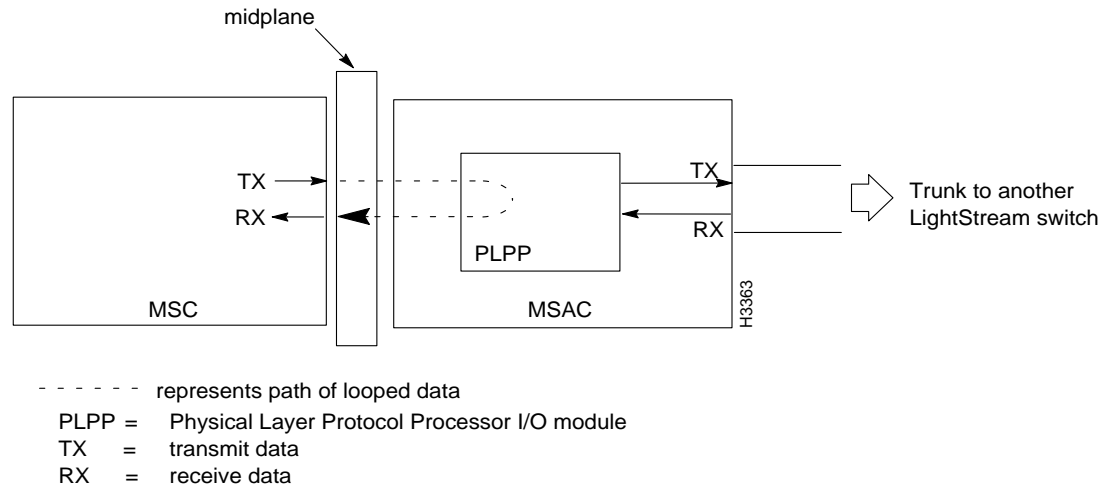
Remote loop failure indicates a problem somewhere between the local access card and the remote system.

Step 3 To run an internal loop on the port, type the following at the cli> prompt:

```
cli> set port <port#> loop internal
```

The CLI automatically sets the administrative status of the selected port to *testing* and starts the loopback test. Figure 7-6 shows how the data is looped during an internal loopback test.

Figure 7-6 An internal loop on an MS trunk port



If the internal loop succeeds, and the local trunk comes up, you have isolated the problem with the local access card.

Step 4 To stop the loopback test, type the following at the cli> prompt:

```
cli> set port <port#> unloop
```

Always use the unloop procedure to stop the loopback test once you obtain the test results, whether or not the test was successful. Leaving a loop up blocks the flow of data and could contribute to congestion on the network.

Step 5 Reset the port to the UNI net interface type by typing the following at the cli> prompt:

```
cli> set port <port#> framerelay netinterfacetype uni
```

Procedure 2: Looping Edge Ports

This procedure describes how to loop data through a frame relay port. The line from the port connects a LightStream switch to an external device provided by another vendor. If you receive an indication that data is not passing between the LightStream switch frame relay port and the host, or that the line is unreliable, use this looping procedure to isolate the problem.

Note Loopback tests do not work on frame relay ports with the LMI type set to UNI.

Step 1 Type the following at the cli> prompt to set the netinterfacetype attribute to NNI:

```
cli> set port <port#> framerelay netinterfacetype nni
```

where

<port#> = The port you want to loop. The port number is in card.port format.

Step 2 To run an external loop on the frame relay port on the LightStream switch, type the following at the cli> prompt:

```
cli> set port <port#> loop external
```

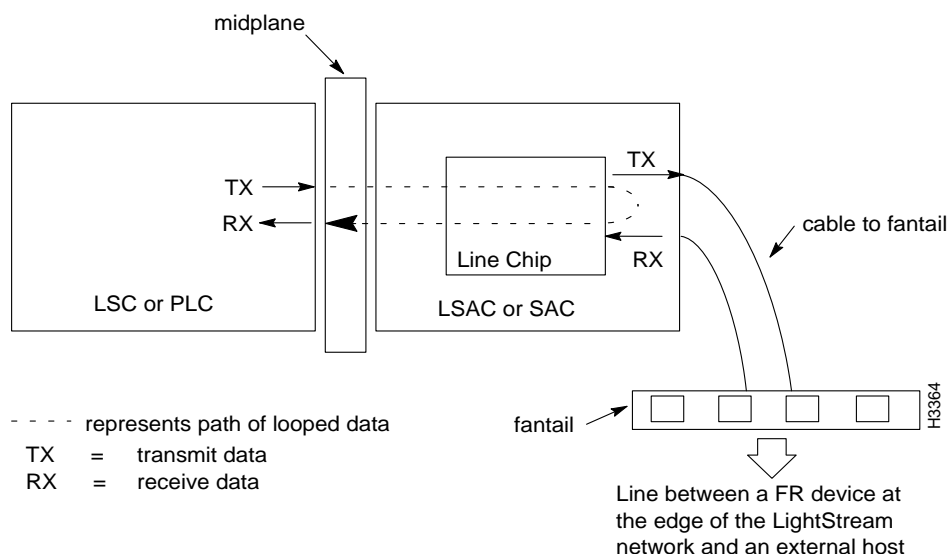

The CLI automatically sets the administrative status of the selected port to *testing* and starts the loopback test.

If the external loop succeeds, the line comes up. A trap stating that the line is up is displayed and the operational status changes to *up*. This means that the local frame relay port is configured correctly and that you have come up in a testing situation. Looping is not used to further isolate the problem.

If the external loop fails (the line does not come up), the frame relay port is not configured properly or a hardware problem may exist. Proceed to Step 3.

Figure 7-7 shows how the data is looped during an external loopback test.

Figure 7-7 An external loop on a frame relay port



Step 3 Run an internal loop on the port by typing the following at the cli> prompt:

```
cli> set port <port#> loop internal
```

where

<port#> = The port you want to loop. The port number is in card.port format (card = 2 - 10; port = 0 - 7).

The CLI automatically sets the administrative status of the selected port to *testing* and starts the loopback test.

If the internal loop succeeds and the line comes up, you have isolated the problem to the SCC I/O module.

Figure 7-8 shows how the data is looped during an internal loopback test.

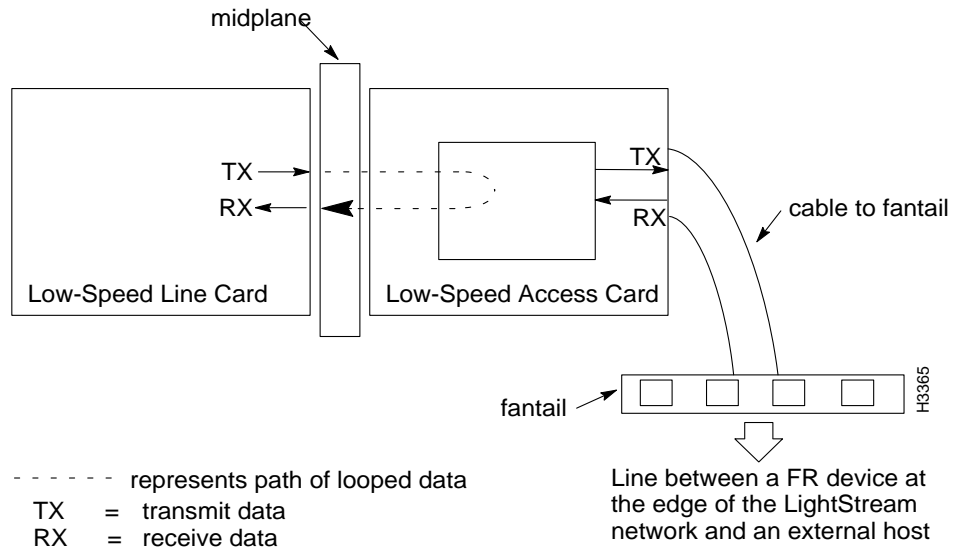
Step 4 To stop the loopback test, type the following at the cli> prompt:

```
cli> set port <port#> unloop
```

where

<port#> = The port you want to unloop. The port number is in card.port format (card = 2 - 10; port = 0 - 7).

Figure 7-8 An internal loop on an LS edge port



Always use the unloop procedure to stop the loopback test once you obtain the test results, whether or not the test was successful. Leaving a loop up blocks the flow of data and could contribute to congestion on the network.

Step 5 Reset the port to the UNI net interface type by typing the following at the cli> prompt:

```
cli> set port <port#> framerelay netinterfacetype uni
```

Other Edge Ports

The procedure for looping other (non-frame relay) edge ports is nearly identical to the procedure for looping frame relay ports except that there is no need to reconfigure the net interface type as in frame relay ports. Therefore, to loop test any other type of edge port, follow the procedure described in "Procedure 2: Looping Edge Ports," disregarding Step 1 and Step 5.

Unlooping Ports

This procedure allows you to unloop the port when you finish a loopback test. Whenever you run a loopback test on a port, you must unloop the port when you finish. However, if you proceed from one type of loopback test to another type on a particular port (from external to remote, for example), you need not unloop the port before you start the next loopback test.

Procedure

To unloop a port, type the following at the cli> prompt:

```
cli> set port <port#> unloop
```

where

<port#> = The port you want to unloop. The port number is in card.port format (card = 2 - 10; port = 0 - 7).

Always use the unloop procedure to stop the loopback test once you obtain the test results, whether or not the test was successful. Leaving a loop up blocks the flow of data and could contribute to congestion on the network.

Expected Results

The administrative status on the port changes from *testing* to *enabled* and the operational status changes from *testing* to either *up* or *down*.

Activating (or Deactivating) a Card

This procedure allows you to set a particular network processor or line card to *active*, *inactive*, or *testing*. If a card is *active*, it is up and passing data. If a card is *inactive*, it is down and cannot pass data. If a card is set to *testing*, it is in diagnostics mode.

Before replacing a component (NP, switch card, line card, or access card), you should deactivate it. After replacing the component, you should activate it.

Procedure

Step 1 Type the following at the cli> prompt:

```
cli> set card <card#> {active|inactive|testing}
```

where

- <card#> = 1 - 10
- {active|inactive|testing} Specifies whether the card is up and able to pass data, down and unable to pass data, or in diagnostics mode.

Step 2 Type the following at the cli> prompt:

```
cli> show card <card#> status
```

Expected Results

The **show card <card#> status** command displays the status of the specified card. Operational Status is the actual status and Administrative Status is the status that you set.

Activating (or Deactivating) a Port

This procedure allows you to set a particular port (except a frame forwarding port) to *active*, *inactive*, or *testing*. If a port is set to *active*, it is up and passing data. If a port is set to *inactive*, it is down and cannot pass data. If a port is set to *testing*, you can set up software loops on the I/O components of the board itself, but not on the DSU/CSU.

Procedure

Step 1 At the cli> prompt, type:

```
cli> set port <port#> {active|inactive|testing}
```

where

- <port#> = The port you want to make active or inactive. The port number is in card.port format (card = 2 - 10; port = 0 - 7).
- {active|inactive|testing} Specifies whether the port is up and able to pass data, down and unable to pass data, or in diagnostics mode. (The default is active.)

Step 2 To determine the status of a particular port, type the following at the cli> prompt:

```
show port <port#> status
```

A screen similar to the following will be displayed.

Figure 7-9 Example – show port command output

```
cli> show port 5001 status
Admin Status:      Up
Oper Status:       Up
Oper loop:         none
Admin loop:        none
Last Oper Change:  25 Hr 2 Min 58 Sec ago
```

H3366

Operational status is the port's actual status and administrative status is the status that you set.

The following are examples of the traps that are displayed when the port goes up or down, respectively:

```
Link up trap from Light7, System Up Time: 23 Hr 29 Min 55 Sec Port: 5.0
Link down trap from Light7, System Up Time: 23 Hr 29 Min 50 Sec Port: 5.0
```

Activating (or Deactivating) a Frame Forwarding Port

This procedure enables you to activate or deactivate a particular frame forwarding port. When you activate a port, a virtual channel connection (VCC) is set up to a designated endpoint and traffic can flow over that connection. When you deactivate a port, no traffic can flow over the connection.

Procedure

At the cli> prompt, type:

```
cli> set port <port#> frameforward {active|inactive}
```

where

- <port#> = The frame forwarding port you want to make active or inactive. The port number is in card.port format (card = 2 - 10; port = 0 - 7).
- {active|inactive} Specifies whether traffic can or cannot flow over the connection. (The default is active.)

Activating (or Deactivating) a Frame Relay DLCI

This procedure describes how to activate or deactivate a particular frame relay DLCI. When you activate a connection, traffic can flow over that connection. When you deactivate a connection, no traffic can flow over it.

Procedure

At the cli> prompt, type:

```
cli> set port <port#> dlci <dlci#> {active|inactive}
```

where

- <port#> = The port that contains the DLCI you want to make active or inactive. The port number is in card.port format (card = 2 - 10; port = 0 - 7).
- <dlci#> = The DLCI you want to make active or inactive. The DLCI number can be between 16 and 991.
- {active|inactive} Specifies whether traffic can or cannot flow over the connection. (The default is active.)

Activating (or Deactivating) an ATM UNI VCI

This procedure enables you to make active or inactive a particular ATM UNI virtual channel identifier (VCI). When a connection is active, traffic can flow over it. When a connection is inactive, no traffic can flow over it.

At the cli> prompt, type:

```
cli> set port <port#> atm-vci <atm vci#> {active|inactive}
```

where

- <port#> = The port that contains the ATM UNI VCI you want to make active or inactive. The port number is in card.port format (card = 2 - 10; port = 0 - 7).
- <atm vci#> = The ATM UNI VCI you want to make active or inactive. The ATM UNI VCI number can be between 1 and 32399.
- {active|inactive} Specifies whether traffic can or cannot flow over the connection. (The default is active.)

Resetting a Card

The procedure describes how to reset a card from the TCS.



Caution Be very careful when resetting the active NP or switch cards. Resetting these cards brings down the entire switch and causes data loss for about 5 minutes on the card.

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 At the *cli> prompt, type:

```
*cli> set tcs <card#> reset
```

where

<card#> = 1 - 10, SA, or SB

Loading Operational Software

This procedure tells you how to load operational software or diagnostics programs into the line cards, network processors, or switch cards. In addition to the operational software, the following diagnostic programs are available:

- Network Processor card
- Low-speed line card (T1)
- Medium-speed line card (T3)

If you load diagnostics into a card, refer to the “Hardware” chapter of the *LightStream 2020 Installation and Troubleshooting Manual* for detailed instructions on how to run the diagnostics and how to interpret the results.

Note The **loadcard** command resets the card before loading any software.

Procedure

Step 1 At the cli> prompt, enter:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 Load the operational software or diagnostics program into the selected card by typing the following at the *cli> prompt:

```
*cli> loadcard <slot#> [<load address>] <file name>
```

where

- <slot#> = 1 - 10
- [<load address>] Specifies the address where the diagnostic program will be loaded. This is an optional argument. If you do not specify this argument, the diagnostic program is loaded into a default load address.
- <file name> Do not enter a file name if you want to load operational software into the card. The file names for the various diagnostic programs are:
 - /usr/diag/diag_np1.aout (NP card diags)
 - /usr/diag/diag_ls1.aout (low speed line card diags)
 - /usr/diag/diag_ms1.aout (medium speed line card diags)

Step 4 If you load a diagnostics program into a card, use the **connect** command to establish a connection to the card you want to test.

```
*cli> connect 1 diagnostic
```

When you finish running diagnostics and want to disconnect from the card and return to CLI, type:

```
~~q
```

Using the ping Command

The **ping** command is used to determine if you can communicate over a particular IP connection. The **ping** command sends an ICMP echo packet to the specified IP address. If IP is working at that address, IP sends an ICMP-echo-reply message to the sender.

Procedure

Step 1 Log in to the root account on the LightStream switch from which you wish to send the ICMP echo packet.

Step 2 Enter the following at the bash prompt:

```
bash# ping [packet size] <host name>
```

where

- [packet size] = The size of the packets sent to the host. This is an optional argument. The default packet size is 64 bytes.
- <host name> = The name or IP address of the host to which packets are sent.

Step 3 To stop **ping** and display a summary of the results, press **^C**.

Expected Results

The information shown in Figure 7-10 is displayed.

Figure 7-10 Example - ping display

```
*cli> ping Light5
PING Light5 (127.1.24.35): 64 data bytes
64 bytes from 127.1.24.35: icmp_seq=0 time=100 ms
64 bytes from 127.1.24.35: icmp_seq=1 time=0 ms
64 bytes from 127.1.24.35: icmp_seq=2 time=0 ms
^C

++++Light5 PING Statistics++++
3 packets transmitted, 3 packets received, 0 packet loss
round-trip (ms)  min/avg/max = 0/33/100
```

H3367

Verifying Software

Follow this procedure to verify that all files and directories that were copied from the installation diskettes to the hard disk are intact. All relevant files used to verify software installation reside on the first diskette of the following installation diskette sets:

- System
- Application
- Firmware
- Diagnostic

Procedure

Step 1 Change to the root (/) directory by typing the following at the bash# prompt:

```
bash# cd /
```

Note Note: You must run ckswinstall from the root account and from the root (/) directory.

Step 2 To verify your software installation, type the following at the bash# prompt:

```
bash# ckswinstall
```

Step 3 Insert the first diskette of the System diskette set. Press **[Return]** when you see the following prompt:

```
Insert the FIRST diskette of the set and press <RETURN>
```

When you press **[Return]**, the information shown in Figure 7-11 is displayed after **ckswinstall** runs on the first set of software.

Figure 7-11 Example - ckswininstall display

```

You inserted diskette 1 of the System set.

Starting verification of installation.

Verifying the System directories.
The System directories have been verified.

Verifying the System files.
The System files have been verified.

Verifying the System hard links.
The System hard links have been verified.

Verifying the System symbolic links.
The System symbolic links have been verified.

Verifying the System special files.
The System special files have been verified.

Finished verification of installation.

```

H3368

Step 4 When you see the message that the verification procedure is complete, repeat this process for the Application, Firmware, and Diagnostic installation diskettes.

Expected Results

Because the software has been running for a period of time when you run **ckswinstall**, some files may have changed from the original installation. The software is probably not corrupted if you receive messages that any of the following directories or files have changed:

- Any files in the /usr/app/base/config directory
- The following .profile files for the standard accounts:
 - /profile
 - /usr/oper/.profile
 - /usr/npadmin/.profile
 - /usr/fldsup/.profile
- Log files in the /usr/etc directory:
 - /usr/etc/ftpdlog
 - /usr/etc/inetdlog
 - /usr/etc/rlogindlog
 - /usr/etc/rshdlog
 - /usr/etc/telnetdlog
 - /usr/etc/tftpdlog
- /.rhosts

- /etc/group
- /etc/motd
- /etc/passwd
- /usr/etc/hosts
- /usr/etc/host.equivalent

If cksinstall detects errors in other directories or files, reinstall the software and repeat this procedure. For software installation instructions, see the “Software” chapter of the *LightStream 2020 Installation and Troubleshooting Manual* and the *LightStream 2020 Release Notes*.

Copying a File Between LightStream Switches

This section tells you how to copy files between different LightStream switches. You can copy files from remote NP disks to local NP disks and vice versa. You may choose to copy files for a number of reasons, such as moving log files from a remote disk to a local disk so you can view them.

You move these files by using the file transfer program from within the CLI **shell** command.

Procedure

Step 1 At the cli> prompt, type:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 At the *cli> prompt, enter:

```
*cli> shell "ftp <IP address or name of the NP you want to copy files to or from>"
```

The LightStream switch responds with output similar to the following:

```
Connected to 127.1.22.25.  
220 emtb5 FTP server (Version 4.162 Tue Nov 1 10:50:37 PST 1988) ready.
```

Step 4 Enter your LightStream login name (oper, npadmin, fldsup, or root) when you see the following prompt:

```
Name (127.1.22.25:oper):
```

Step 5 Enter the LightStream password for this account on the destination NP when you see the following prompt:

```
331 Password required for oper.  
Password:
```

If you entered the login name and password correctly, the LightStream switch displays the following information:

```
230 User oper logged in.  
ftp>
```

Step 6 Enter the following at the ftp> prompt:

```
ftp> put /usr/tmp/mma/mma.traplog [<new name>]
```

where

<new name> = The name of the file that identifies the chassis or appropriate directory name for the file. For example, if you are copying a trap log for a switch called Light5, the new name could be `mma_Light5.traplog`.

This command sends the log file to the specified workstation or host. The system tells you when the copy is complete.

Step 7 To exit the file transfer program and return to CLI when you are finished, type:

```
ftp> bye
```

or

```
ftp> quit
```

Step 8 To verify that the files and directories were transferred correctly, enter the following at the `*cli>` prompt:

```
*cli> shell "ls -l"
```

Expected Results

The files and directories are copied to the specified switch.

Turning Power On and Off in a Slot

This section tells you how to turn on or off the +5 V (VCC) power supply for a card in a given slot. When you insert a new card, the power comes on automatically. This procedure affects the power to both the function card and its access card.



Caution Always use this procedure to turn the power off before you remove a line card or its associated access card from the chassis. Removing cards with the power on could result in damage to your hardware.

Procedure

At the `cli>` prompt, enter:

```
cli> set tcs <card#> power {on|off}
```

where

- <card#> = SA, SB (for switch cards) or 1 - 10 (for NP or line cards)
- {on|off} Indicates whether you are turning the power on or off.

Operational Tips

Turning on the power in a slot does not bring up a card in its initialized state. Refer to the *LightStream 2020 Operations Guide* for instructions on bringing the card up.

Diagnosing Port Problems

This section tells you how to diagnose port problems. The procedures (to be performed in sequence) described below include:

- Performing basic port checks
- Checking bit rates
- Checking connections
- Checking receive data
- Checking line statistics
- Checking CSU statistics
- Checking LSC trunks
- Diagnosing VC problems
- Tracing data

Performing Basic Port Checks

This procedure contains the basic port checks and verifies that the port is enabled. See Figure 7-12 for an example of the statistics displayed.

Procedure

- Step 1** To display port information and ensure that the port is enabled, type the following at the cli> prompt:

```
cli> show port <port#> all
```

where

<port#> = The port number for which information will be displayed. The port number is in card.port format (card = 2 - 10; port = 0 - 1).

A screen similar to Figure 7-11 is displayed.

- Step 2** Check the receive data entry for excessive line errors, dropped packets, or the lack of receive data.
- Step 3** Check the remaining line statistics for excessive line errors, dropped packets, and lack of receive data.
- Step 4** Check that the Admin Status entry to is set to *up*.
- Step 5** If the port is an LSC port, check the Measured Bit Rate. (Refer to the section “Checking Bit Rates.”)

If the port is an MSC port, check the MSC port configuration parameters. (Refer to the section “MS Configuration Parameters.”)

Figure 7-12 Example - show port all display

```

cli> show port 5001 all
Description:                Low Speed Edge Line Card Rev 1.0
Port Name:                  Light8.5.1_fr
Port Type:                  LS Edge
MIB2 Type:                  ds1
Port MTU:                   1516 Octets
Port Speed:                 512000 bps

Admin Status:               Up
Oper Status:                Up
Oper loop:                  none
Admin loop:                 none
Last Oper Change:           23 Hr 28 Min 2 Sec ago

Octets Rcvd:                 1280586
Normal Packets Rcvd:         24162
Multicast Packets Rcvd:      0
Discarded Rcvd Packets:      1
Receive Errors:              4
Unknown Protocols Rcvd:      0
Octets Sent:                 1329505
Normal Packets Sent:          25085
Multicast Packets Sent:       0
Discarded Output Packets:     4389
Output Errors:               12

Port Type:                   v35
Oper CSU Type:               none
Admin CSU Type:              none
Oper DCE Rcv Bit Rate:       512000 bps
Admin DCE Rcv Bit Rate:      512000 bps
Oper DCE Xmit Bit Rate:       512000 bps
Measured Bit Rate:           512100 bps
Link Transmit Utilization    1 cells/sec
Admin Expected DTE Rate:     896000 bps
Oper Net Interface Type:     dce
Admin Net Interface Type:    dce
Oper Protocol:                MS Trunk
Admin Protocol:               MS Trunk

LC Auto Enable State:        Disabled
LC Debug Level:              0
Port Data Cell Capacity:     1163 cells
Port Available Capacity:     1163 cells

```

(Continued)

Figure 7-13 Example - show port all display (concluded)

```
Call Setup Retry Time:      5
Call Setup Backoff Time:   5
Oper Max Frame Size:       1516
Modem Status:              RTS:1  DTR:1

Src Node:                   lstb5
Src Port:                   3.6
Dest Admin Node:           lstb6
Dest Operational Node:     lstb6
Dest Admin Port:           6.7
Dest Operational Port:     6.7

Src Admin Insured Rate:    384000 bps
Src Oper Insured Rate:     383966 bps
Src Admin Insured Burst:   6064 bytes
Src Oper Insured Burst:    6052 bytes
Src Admin Max Rate:        512000 bps
Src Oper Max Rate:         511841 bps
Src Admin Max Burst:       12128 bytes
Src Oper Max Burst:        12105 bytes

Dest Oper Insured Rate:    0 bps
Dest Oper Insured Burst:   0 bytes
Dest Oper Max Rate:        0 bps
Dest Oper Max Burst:       0 bytes

To±Net Circuit ID:         0
To±Net Circuit State:      Inactive
From±Net Circuit ID:       0
From±Net Circuit State:    Inactive
Last ATMM Error:           ATMM error 51:Not enough inbound bandwidth
Cells Required:1219

CLP=0 Frames to Switch:    0
CLP=0 Cells to Switch:     0
CLP=1 Frames to Switch:    0
CLP=1 Cells to Switch:     0
Discarded Frames:0
Discarded Cells:           0
CLP=0 Frames from Switch:  0
CLP=0 Cells from Switch:   0
CLP=1 Frames from Switch:  0
CLP=1 Cells from Switch:   0
```

H3370

Checking Bit Rates

This procedure allows you to determine if the cable type is correct and if the device is providing clock. After you ensure that the port is enabled, check the Measured Bit Rate as indicated in the procedure below.

Procedure

Step 1 To display the port information, type the following at the cli> prompt:

```
cli> show port <port#> all
```

where

<port#> = The port number for which information will be displayed. The port number is in card.port format (card = 2 - 10; port = 0 - 1).

A display similar to Figure 7-12 will be displayed.

Step 2 Review the Measured Bit Rate to ensure that it is legal.

Step 3 Compare the Measured Bit Rate with the Admin DCE Rcv Bit Rate. If the Measured Bit Rate is significantly different from the Admin DCE Bit Rate, a problem exists.

If the LS port is set as the DCE, it provides clock. If it is set as DTE, it uses clocking supplied by the attached device (CSU/DSU, router).

A DCE provides a set of clock rates limited by the clock crystals it has available. If an attempt is made to activate an invalid bit rate, the following trap is displayed:

```
(INFO) LCC_3037 at 19.28/94 14:27:52 EDT (10/28/94 18:27:52 GMT) LCC port 7000 unable
to source clock at 70001 bits/second.
```

In this case, the Admin DCE Bit Rate will not equal the Oper DCE Bit Rate. Otherwise, if the port is a DTE, one of several problems could exist. Since the correct clock is not being detected, one of two problems may exist.

- The wrong cable type used to connect the local LightStream port and the device.
- The device being connected to is not providing clock.

Step 4 If the Measured Bit Rate is correct, a problem may still exist. Check to see if there is a total lack of receive data or a very high Receive Errors rate. Make a note of the Octets Rcvd and Normal Packets Rcvd entries.

If the LightStream port is configured as DCE, the measured bit rate simply shows that the port is correctly generating transmit clock. Receive clock is provided by the host or external trunk, which must reflect the transmit clock. (Some vendors do not adhere to the physical layer requirement in this regard.) This situation is marked by a total lack of receive data or a very high Receive Errors rate on the receive data.

Step 5 If there is a total lack of receive data or a very high error rate on the receive data, review the physical layer connection.

MS Configuration Parameters

On a medium-speed line card (MSC), a high error rate or a total lack of errors with data transfer not working can be caused by an incorrect setting of the cell payload scrambling attribute, the DS3 line type, or the cable length attributes. The next sections describe two parameter mismatches and some of their symptoms.

Note Cell payload scrambling appears as PLCP scrambling in some displays.

Mismatched Cell Payload Scrambling

On MSC trunks, cell payload scrambling is disabled by default. If one end of a trunk has cell payload scrambling *enabled* and the other end has cell payload scrambling *disabled*, packets will appear to be received and transmitted without error in the port statistics display. The trunks will never come up, however, because the payload of the cells is scrambled at one end and not unscrambled at the other end. This causes the TUD protocol to fail.

A UNI port will also appear to function normally without transmit or receive errors. However, the hosts that eventually try to reassemble the cells into useful data will find that the data is corrupted. A series of AAL5 CRC errors occur if the calls carry AAL5-encoded data.

Incorrect Line Type

If the Line Type parameter is set incorrectly (dsx3Linetype mismatch), a very low error rate will appear. An idle trunk will regularly count a receive error every 3 to 5 seconds.

Clocking Checks

To check clocking, review the connections and check the statistics as shown in the procedures below.

Checking Connections

To check connections, follow the procedure below:

Procedure

Step 1 If LightStream ports are directly connected to a host, ensure that one side is configured as a DCE and that the other side is configured as a DTE.

Step 2 If the LightStream ports are connected through CSUs, ensure that both ports are configured as DTEs.

Step 3 To display the port information, type the following at the cli> prompt:

```
cli> show port <port#> all
```

where

<port#> = The port number for which information will be displayed. The port number is in card.port format (card = 2 - 10; port = 0 - 1).

A screen similar to Figure 7-12 will be displayed.

If a port is configured as a DTE, the clock rate being received and used to transmit is shown as the Measured Bit Rate.

Step 4 Check the Measured Bit Rate. If it is not similar to the Admin DCE Bit Rate and the port is correctly configured as a DTE, a hardware fault or cable problem exists. Run diagnostics on the port. (Refer to the “Hardware” chapter of the *LightStream 2020 Installation and Troubleshooting Manual* for detailed instructions on how to run the diagnostics and how to interpret the results.)

Checking Receive Data

To check the receive data, follow the procedure below:

Procedure

- Step 1** To display the line statistics, type the following at the cli> prompt:
- ```
cli> show port <port#>
```
- Step 2** Note the Octets Rcvd and Normal Packets Rcvd statistics as shown below:
- ```
cli> show port 5001
Octets Rcvd: 588815531
Normal Packets Rcvd: 11109727
```
- Step 3** To determine if the statistics are increasing, repeat the command by typing the following at the cli> prompt:
- ```
cli> show port <port#>
```
- Step 4** Review the Octets Rcvd and Normal Packets Rcvd statistics. (See the changes in the statistics shown below.) If the statistics increase, the port is receiving data.
- ```
cli> show port 5001
Octets Rcvd: 588857719
Normal Packets Rcvd: 11110540
```
- Step 5** If the Octets Rcvd and Normal Packets Rcvd statistics do not increase, the port is not receiving data. Check clocking and cabling.

Checking Line Statistics

To check line statistics, follow the procedure below.

Procedure

- Step 1** To display the line statistics, type the following at the cli> prompt:
- ```
cli> show port <port#>
```
- Step 2** Note the Octets Rcvd and Normal Packets Rcvd statistics as shown in below.
- ```
cli> show port 5001
Octets Rcvd: 588857719
Normal Packets Rcvd: 11110540
```
- Step 3** To determine if the statistics are increasing, repeat the command by typing the following at the cli> prompt:
- ```
cli> show port <port#>
```
- Step 4** Review the Octets Rcvd and Normal Packets Rcvd statistics. If the statistics increase, the host or remote trunk is sending data.
- Step 5** If the Octets Rcvd and Normal Packets Rcvd statistics do not increase, refer to the sections “Performing Basic Port Checks,” “Checking Bit Rates,” and “Clocking Checks.”
- Step 6** Review the Discarded Rcvd Packets statistic. If no VCs exist to carry the incoming data or if that data is being dropped by the UPC code for the given VCs, the Discarded Rcvd Packets statistic will increase. Refer to the section “Diagnosing VC Problems.”
- Step 7** Review the Receive Errors statistic. If the incoming data is being received incorrectly, the Receive Errors statistic will increase. If the statistic is increasing, refer to the sections “Performing Basic Port Checks,” “Checking Bit Rates,” and “Clocking Checks.” If CSUs are in use, review the CSU statistics for that CSU.

- Step 8** Review the Octets Sent and Normal Packets Sent statistics. If data is being sent out from a port, the Octets Sent and Normal Packets Sent statistics increase. This should always be the case for a trunk, and it should also occur with FR if NNI is selected with a non-null LMI type.
- Step 9** Review the Output Errors statistic. If there is a problem transmitting data, the Output Errors statistic will increase.

### Checking CSU Statistics

The LSC CSU statistics are available only by connecting to the CSU through a serial line. If the CSU is connected to the DSU/CSU control port, the CSU can be reached with the **csumon** command from the Lynx shell. The CSU statistics for the MSC are available using the standard DS3 MIB variables; some can also be displayed using csumon. The CSU statistics for the CLC/OC3 card are available using the SONET MIB. See the *LightStream 2020 Operations Guide* for information on using the csumon utility to monitor DSU/CSU statistics.

### Checking LSC Trunks

If the trunk does not come up or the frame relay port does not come up, use the procedures outlined in the sections “Symptom - Trunk Will Not Come Up,” and “Symptom - Frame Relay Port Does Not Come Up,” respectively, to determine the cause of the problem.

### Symptom - Trunk Will Not Come Up

If the trunk will not come up, perform the checks shown in the procedure below.

#### Procedure

- Step 1** To display the port information, type the following at the cli> prompt:
- ```
cli> show port <port#> statistics
```

Figure 7-14 Example - show port statistics display

```
cli> show port 5001 statistics

Octets Rcvd:                234061260
Normal Packets Rcvd:        2949444229
Multicast Packets Rcvd:     0
Discarded Rcvd Packets:    107
Receive Errors:             56702
Unknown Protocols Rcvd:    0
Octets Sent:                1068586222
Normal Packets Sent:        2046089974
Multicast Packets Sent:     0
Discarded Output Packets:   8490
Output Errors:              12
```

H3374

- Step 2** Check the port at each end of the trunk with the command shown in Step 1. Make sure that both ports are periodically sending cells.
- Step 3** Review the Octets Sent statistic to verify that it is increasing.
- Step 4** If one side of the trunk is not sending, make sure it is enabled as a trunk. Refer to the section “Performing Basic Port Checks.”
- Step 5** If one port never sends Trunk-Up-Down messages, make sure the card is correctly configured as a trunk card.

If port 0 is not configured on a card, the card type as configured in the line card EEPROM determines the operational type of the card. In this case, the configuration values on the remaining ports are rejected and a trap is generated.

To solve this problem, configure a trunk on port 0. You can configure it as *inactive*, if desired. When the configuration is downloaded, the line card EEPROM will be updated and the line card and LCC process will both be restarted.

- Step 6** If both sides of the trunk show that they are sending cells, find out which side is not receiving cells. Follow the steps in the section “Performing Basic Port Checks” to determine why the trunk ports cannot communicate.

If a frame relay port does not come up (the administrative status is *up*, but the operational status is *down*), follow the steps in the procedure below.

Symptom - Frame Relay Port Does Not Come Up

Procedure

- Step 1** Follow the steps in the section “Performing Basic Port Checks.”
- Step 2** Make sure that both the frame relay DCE and the frame relay host are configured to use the same LMI protocol. Both must use FRIF, ANSI or ITU/TSS.

Step 3 Make sure that the LightStream port is correctly configured as a UNI port or NNI port. A UNI protocol should usually be used as the NNI protocol is designed for network device to network device connection and is rarely used.

Step 4 To display the port information, type the following at the cli> prompt:

```
cli> show port <port#> all
```

Step 5 Check that the Normal Packets Received statistic is increasing. A packet should be received every 10 seconds. The FR host is responsible for sending a status enquiry at periodic intervals. (The default for this interval is every 10 seconds.) If the Normal Packets Received statistic is increasing, continue with Step 7.

Step 6 Review the Discarded Received Packets statistic. If the Discarded Received Packets entry is increasing, the packets are coming in, but on a different DLCI.

This will occur when the incorrect LMI type is selected. (The FRIF LMI DLCI is 1023. The ANSI and ITU/TSS LMIs use DLCI 0.) Check these settings again or test by switching the LMI type to see if the FR port becomes active.

Step 7 If the LMI does not come up, make sure that packets are being received on the LMI DLCI.

Each time a frame is received with the FR LMI DLCI, the MIB variable frameRelayDlciToSwCLP0Frames.port.lmidlci should increase. For example, using an ANSI LMI on card 7 port 4, read the MIB variable frameRelayDlciToSwCLP0Frames.7004.0. If that variable is not readable, it means the port is not configured as an active FR port with that same LMI. Review the MIB variables to be sure.

Step 8 If the frameRelayDlciToSwCLP0Frames increases, check to see that it is being processed correctly. You may need more trap information from the FR DCE port. Proceed to Step 9.

If the frameRelayDlciToSwCLP0Frames variable is not increasing, the frames are not being received with the correct DLCI.

Step 9 To determine if the LCC is having trouble receiving or interpreting the LMI packets, type the following at the cli> prompt:

```
cli> set trap "LCC_3*"
```

If the LCC-based LMI module is enabled and is not responding to STATUS_ENQUIRY messages, an LCC trap should be seen periodically. (The default is every 10 seconds.) Some of the potential LCC traps, their meanings and some possible or required actions are listed in Table 7-1.

Table 7-1 LCC Traps Text, Definitions, and Actions

Trap Text	Meaning/Action
LCC_3000 "lccFrLmiSendVc write error portid %d"	Report this problem to LightStream Corp.
LCC_3008 "Frame Relay Port %d - data unused dlci %d"	An LMI type mismatch exists. If the dlci reported is 1023, the host is sending FRIF frames and the frame relay edge port is configured for ANSI or ITU/TSS. If the dlci reported 0, the host is sending ANSI or ITU/TSS frames to a FR edge port configured for FRIF.
LCC_3013 "Frame Relay Port %d - LMI unknown IE %d at offset %d"	This is a typical trap when the FR host is configured as ANSI and the FR edge port is configured as ITU/TSS.
LCC_3017 "Frame Relay Port %d - LMI missing mandatory IE, mask=%x"	This is a typical trap when the FR host is configured as ANSI and the FR edge port is configured as ITU/TSS.

Trap Text	Meaning/Action
LCC_3027 "Frame Relay Port %d - LMI missing lockshift5"	This trap is typical when the FR host is configured as ITU/TSS. and the FR edge port is configured as an ANSI LMI.
LCC_3005 "Frame Relay Port %d - frame too small"	This trap indicates serious incompatibilities or corruption of data.
LCC_3006 "Frame Relay Port %d - EA bit incorrect"	This trap indicates serious incompatibilities or corruption of data.
LCC_3007 "Frame Relay Port %d - data on unstarted dlci %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3009 "Frame Relay Port %d - Frame too small, size %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3010 "Frame Relay Port %d - Invalid LMI header at offset %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3011 "Frame Relay Port %d - Invalid LMI message type %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3012 "Frame Relay Port %d - LMI IE Truncated at offset %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3014 "Frame Relay Port %d - Repeated LMI mandatory IE %d at offset %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3015 "Frame Relay Port %d - LMI frame contains wrong IE %d at offset %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3016 "Frame Relay Port %d - LMI message contains %d excess bytes"	This trap indicates serious incompatibilities or corruption of data.
LCC_3018 "Frame Relay Port %d - LMI received invalid report type %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3019 "Frame Relay Port %d - LMI frame contained invalid PVC DLCI %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3020 "Frame Relay Port %d - LMI frame reported too many PVCs"	This trap indicates serious incompatibilities or corruption of data.
LCC_3021 "Frame Relay Port %d - LMI frame received with DLCIs misordered"	This trap indicates serious incompatibilities or corruption of data.
LCC_3022 "Frame Relay Port %d - LMI expected report type %d, got %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3026 "Frame Relay Port %d - LMI out of buffers to send message"	This trap indicates serious incompatibilities or corruption of data.
LCC_3028 "Frame Relay Port %d - LMI frame contained invalid PVC status"	This trap indicates serious incompatibilities or corruption of data.
LCC_3029 "Frame Relay Port %d - Provider Primitives invalid/incomplete"	This trap indicates serious incompatibilities or corruption of data.
LCC_3030 "Frame Relay Port %d - LMI bad IE length %d at offset %d"	This trap indicates serious incompatibilities or corruption of data.
LCC_3025 "Frame Relay Port %d - LMI"	Resource shortage or a misconfiguration of FR to allocate fewer VCs than it needed
LCC_3036 "FR Port %d LMI timer expired without required message, lmi dlci %d, user (1)/net(2) %d"	The local LMI will send this trap if it is not receiving valid STATUS_ENQUIRY messages or if an NNI is not receiving valid STATUS messages.
LCC_3023 "Frame Relay Port %d - LMI sequence number mismatch, expected %d received %d"	Some messages are being lost between the edge port and the host.
LCC_3024 "Frame Relay Port %d - LMI reply is not for last enquiry, current %d last %d"	Some messages are being lost between the edge port and the host.

Diagnosing VC Problems

If a FR, FF, or UNI VC is not created or if no data is being delivered over a FR VC, perform the procedures in this section.

Procedure: A FR, FF, or UNI VC Fails to Be Created

- Step 1** Verify that the VC is configured on both end points with the CLI **show port <port#>** command. If one end point is missing, is inactive, or is on an inactive port, the VC will not be created.
- Step 2** Look at the explanation in the Last ATM Error variable for that VC. (Refer to Figure 7-12.) The most common errors, their definitions, and user actions are shown in Table 7-2.
- If no data is being delivered over a FR VC, perform the checks outlined in the section “Procedure: No Data Being Delivered over a Frame Relay VC.”

Procedure: No Data Being Delivered over a Frame Relay VC

- Step 1** To check the VC status, type the following at the cli> prompt:
- ```
cli> show port <port#> dlci <dlci#>
```
- where
- <port#> = The port that contains the DLCI you want to view. The port number is in ccppp or card.port format (card = 2 - 10; port = 0 - 7).
  - <dlci#> = The DLCI you want to view. The DLCI number can be between 16 and 991.
- Step 2** If the VC has been established, review the statistics for the VC by typing the following at the cli> prompt:
- ```
cli> walksnmp lsFrameRelayDlciStatTable
```

Table 7-2 lastAtmError Text, Definitions, and Actions

Error Text	Definition	What to Do
No flow resources	The cardMaxVCs attribute is set too low.	Increase this value and reboot that line card.
No connect resources	The cardMaxVCs attribute is set too low.	Increase this value and reboot that line card.
Invalid connect request	The VC has illegal attributes set.	Review the bandwidth values in particular. A VC cannot have a MaxRate larger than the port. Also, certain combinations of parameters are illegal. If a VC uses “Guaranteed” bandwidth, it is not allowed to have any excess bandwidth; the insured rate must equal the max rate.
Unknown destination	The local node is out of communication with the destination node.	Check to see if any trunks are down that are supposed to connect the nodes.
Not enough outbound bandwidth	No path exists with sufficient bandwidth to support the VC.	Review the VC attributes. The Cells Required attribute shows how many cells worth of bandwidth are needed to carry that VC over a trunk.

Error Text	Definition	What to Do
No path	No path exists with sufficient bandwidth to support the VC.	Review the VC attributes. The Cells Required attributes shows how many cells worth of bandwidth are needed to carry that VC over a trunk.
Destination refused connection	The remote end of the VC is refusing the connection.	Review the VC settings on the remote end point.
Not enough inbound bandwidth	Not enough bandwidth exists on the source port for this VC.	Check the Cells Required attributes to determine the bandwidth required to carry that VC over the source port. Check the Cells Available attribute to determine the bandwidth that has not been allocated to other VCs.

Symptom - Partial Data is Being Delivered over the FR, FF, or UNI

If partial data is being delivered over the FR, FF, or UNI VC, check to see if the network is congested.

Tracing Data

Incoming Edge

On a LSC FR Edge port, you must first determine the VCI of the VC you wish to monitor. You can do this under lmon using the **dlci <port#> <dlci#>** command to determine the flowid.

Step 1 For example, to check port 4 dlci 0 (the ANSI LMI on port 4), enter the following at the prompt:

```
dlci 4 0
```

Step 2 In addition, to get frCktInfoCallIDIncoming and frCktInfoCallIDOutgoing (.port.dlci), type the following at the cli> prompt:

```
cli> getsnmp frCktInfoCallIDIncoming.port.0
```

or

```
cli> getsnmp frCktInfoCallIDOutgoing.port.0
```

Step 3 On an LSC FF Edge port, you must first get the VCI by typing the following at the cli> prompt:

```
cli> getsnmp ffCktInfoCallIDIncoming.port.0
```

or

```
cli> getsnmp ffCktInfoCallIDOutgoing.port.0
```

Step 4 Once you have the VCIs, you can trace them on the line card with the lmon command shown below:

```
trace <port#> <FSU_ID TSU_ID>
```

where

<FSU> = from switch unit (and out the port)

<TSU> = to switch unit (and to the card the cells will be delivered)

Note Note: This release has incoming and outgoing reversed.

The example in Figure 7-15 shows a trace on port 3 of a FF flow that is carried in on VCI 7 and out of the port on VCI 9.

Figure 7-15 Trace example

```
LC_&) trace 3 9 7

PP:OUT(ff) port 3 size 48 ID 9      <0001 0308 0075 5101 0153 028c 9b1c 6a9d>
PP:                                <0000 0000 0000 0000 0000 0000 0000 0000>
PP:                                <0000 0000 0000 0000 0000 0000 8cc9 25fc>
PP: IN(ff) port 3 size 13 ID 7      <0001 0308 007d 5101 0153 029c 8c0f>
PP: IN(ff) port 3 size 13 ID 7      <0001 0308 0075 5101 0053 0252 8bf2>
PP: OUT(ff) port 3 size 48 ID 9      <0001 0308 007d 5101 0053 028c 5257 0301>
PP:                                <c882 5703 06a0 829d a001 0000 0000 0000>
PP:                                <0000 0000 0000 0000 0000 0017 e916 1366>
PP: OUT(ff) port 3 size 48 ID 9      <0001 0308 0075 5101 0153 028d 9c7b 0700>
PP:                                <0000 0000 0000 0000 0000 0000 0000 0000>
PP:                                <0000 0000 0000 0000 0000 0000 9bad a3d1>
PP: IN(ff) port 3 size 13 ID 7      <0001 0308 007d 5101 0153 029d 8d5e>
PP: IN(ff) port 3 size 13 ID 7      <0001 0308 0075 5101 0153 0253 8cd1>
PP: OUT(ff) port 3 size 48 ID 9      <0001 0308 007d 5101 0153 028d 533c 149d>
PP:                                <0000 0000 0000 0000 0000 0000 0000 0000>
PP:                                <0000 0000 0000 0000 0000 000d 79b8 e7c5>
```

H3375

Step 5 Once you have an incoming TSU VCI, you can use it to get more information on the VC, as shown in Figure 7-16, by typing the following at the prompt:

```
vci <vci#>
```

Figure 7-16 FF VC destination slot and FSU priority example

```
LC_7) vci 7
word0 0x0 MBID 7 HQ 0 DGQ 2
SwPri 1 Slot 7 FSUPri 2 XCLP 0 OutPort 10
VPI 0 VCI 8 PT 0 MC 0
TSU free cell count: 0x1ffa
```

H3376

This shows information on the FF VC that uses VCI 7. Among the information shown is the destination slot (7) and FSU priority.

NP-based Tracing

If you can determine the VCIs used on the NP to send and receive data, you can trace the data on those VCs.

Perform the trace by typing the following at the cli> prompt:

```
cli> vcitap <vci#>
```

Figure 7-17 NP-based trace example

```
vcitap 1023
PROGRAM: vcitap: compiled Oct 21 1994 @ 05:27:14 in /nfs/dungeon/u5/lights-
tream.sweng/eml.1/lynx/build/src/cmd/tests/vcitap
Monitoring traffic on VCI 23
SND:(23,len=40):1001 0384 0014 1903 0105 03b2 01cd 0000 0000 0053 0b00 0000 0000
0000 0000 0000 0000 0000 0000
SND:(23,len=40):1001 0384 0014 1903 0105 03b3 01cd 0000 0000 0053 0b00 0000 0000
0000 0000 0000 0000 0000 0000
SND:(23,len=40):1001 0384 0014 1903 0105 03b4 01ce 0000 0000 0053 0b00 0000 0000
0000 0000 0000 0000 0000 0000
SND:(23,len=40):1001 0384 0014 1903 0105 03b5 01cf 0000 0000 0053 0b00 0000 0000
0000 0000 0000 0000 0000 0000
SND:(23,len=40):1001 0384 0014 1903 0105 03b6 01d0 0000 0000 0053 0b00 0000 0000
0000 0000 0000 0000 0000 0000
```

H3377

Optimizing the Load Across Trunks

Using the Trunkmon Program • Optimizing the Load Across Trunks

This chapter tells you how to determine the status of each trunk port in the LightStream® network and the connections that are routed through each trunk. It also provides a procedure to manually reroute frame forwarding, frame relay, and ATM UNI connections so that you can optimize the load across the trunks in your network. These procedures include:

- Using the trunkmon program to view the state of all trunks in the network and determine which connections run through those trunks.
- Taking down connections manually and allowing them to be rerouted through the network so that the load can be optimized across trunks.

All trunk connections are manually configured. After you configure the endpoints and the type of service you want, the LightStream network automatically selects the best path through the network for each connection.

If a trunk fails, a trap is recorded in the trap log, and the LightStream network automatically reroutes the connection, if a path exists between the source and destination ports and if adequate bandwidth is available. When the failed trunk is restored, you must manually reroute the connections back to it. The LightStream switch does not have the ability to automatically shift the connection back to the original trunk.

Using The trunkmon program

The trunkmon program allows you to look at individual trunk ports in the network and determine the state of the trunk, the virtual channel connections (VCCs) that are routed over the trunk, and the state of those VCCs. Before attempting to reroute any of VCCs in the network to balance the load across all available trunks, use the trunkmon program to determine the routes of each VCC.

A major differences between the information provided by the trunkmon program and the command line interface (CLI) commands **show chassis listdlci** and **show chassis listvci** is that the trunkmon program shows you every connection that passes through a particular trunk port, even if the connection does not originate in that chassis. Only the **show chassis listdlci** or **show chassis listvci** commands display information on the VCCs that originate from the specified chassis.

If you use the trunkmon program periodically to become familiar with the way connections are routed through your network, you will know the connections that should be rerouted after a failed trunk has been restored.

The trunkmon program provides a number of commands that can be used to display specific information. The commands available to determine which connections should be rerouted are shown in Table 8-1.

Table 8-1 Trunkmon Commands and Functions

trunkmon Command ¹	Description/Function
help	Displays information describing the different options that are available.
list	Lists all configured trunk ports on the local chassis.
state	Displays summary information for the specified trunk port. This option is often used to focus on a particular trunk port after using the list option.
entry	Displays the source (edge port) of each connection that enters the specified trunk port.
exit	Displays the source (edge port) of each connection that exits from the specified trunk port.
quit	Exits the trunkmon program.

1. The trunkmon program has a number of other commands that are not described here. Only the commands related to optimizing the load across all trunks are discussed in this guide.

You can run the trunkmon program from either the LynxOS shell or from the shell command within CLI. The following two procedures show how to use each of these options.

Procedure 1: Running the trunkmon Program from the bash Prompt

- Step 1** Log in to the fldsup account or the root account for the LightStream switch.
- Step 2** Enter the following at the bash\$ prompt:

```
bash$ trunkmon
```

The trunkmon program is opened and the system displays the trunk> prompt.
- Step 3** For information on the commands provided by the trunkmon program, enter the following at the trunk> prompt:

```
trunk> help {<command name>|all}
```

where

<command name> = The name of one of the commands described above. If you enter **all** instead of a command name, the trunkmon program displays a complete list of all options available.
- Step 4** To use the **list** command, enter the following at the trunk> prompt:

```
trunk> list
```
- Step 5** To use either the **state**, **entry**, or **exit** commands, type:

```
trunk> state <trunk port>
trunk> entry <trunk port>
trunk> exit <trunk port>
```

respectively, where

<trunk port> = The trunk port on which you want information. The trunk port is in the form:

<chassis>.<card #>.<port #> or <chassis>:<card #>.<port #>

where

<chassis> = The chassis ID or chassis name attribute of the chassis containing the trunk port of interest.

<card #> = The number of the card containing the trunk port of interest. The value is between 2 and 10.

<port #> = The port number of interest. The value is between 0 and 7 for low speed trunks and 0 and 1 for medium speed trunks.

For example, to display the state of trunk port 1 on card 3 of the LightStream switch with a chassis ID of 5242 and a chassis name of boston, you would enter either **state 5242.3.1** or **state boston.3.1**.

Step 6 To exit the trunkmon program, enter the following at the trunk> prompt:

```
trunk> quit
```

Procedure 2: Running the trunkmon Program from the CLI

Step 1 At the cli> prompt, enter the following at the cli> prompt:

```
cli> protected
```

Step 2 Enter the protected mode password when you see the following prompt:

```
Enter password:
```

Step 3 To run the trunkmon program from the CLI protected mode, enter the following at the *cli> prompt:

```
*cli> shell "trunkmon"
```

The trunkmon program is opened and the system displays the trunk> prompt.

Step 4 For information on the commands provided by the trunkmon program, enter the following at the trunk> prompt:

```
trunk> help {<command name>|all}
```

where

<command name> = The name of one of the commands described above. If you enter all instead of a command name, the trunkmon program displays a complete list of all options available.

Step 5 To use the **list** command, enter the following at the trunk> prompt:

```
trunk> list
```

Step 6 To use either the **state**, **entry**, or **exit** commands, type:

```
trunk> state <trunk port>
trunk> entry <trunk port>
trunk> exit <trunk port>
```

respectively, where

<trunk port> = The trunk port on which you want information. The trunk port is of the form:

<chassis>.<card #>.<port #> or <chassis>:<card #>.<port #>

where

<chassis> = The chassis ID or chassis name attribute of the chassis containing the trunk port of interest.

<card #> = The number of the card containing the trunk port of interest. Value is between 2 and 10.

<port #> = The port number of interest. Value is between 0 and 7 for low speed trunks and 0 and 1 for medium speed trunks.

For example, to display the state of trunk port 1 on card 3 of the LightStream switch with a chassis ID of 5242 and a chassis name of boston, you would enter either **state 5242.3.1** or **state boston.3.1**.

Step 7 To exit the trunkmon program, enter the following at the trunk> prompt:

```
trunk> quit
```

Expected Results

Figure is an example of output from the trunkmon program.

Figure 8-1 Output from the trunkmon program

```

*cli> shell "trunkmon"

LightStream Trunk Monitor program
PROGRAM: trunkmon: compiled Aug 28 1994 @ 04:39:33 [pid:31]

trunk> help all
state <Port EIA>
    - Display summary state information for trunk <Port EIA>.
check <Port EIA>
    - Check for entry VCC consistency across trunk <Port EIA>.
dlevel <Card EIA> <Debug Level>
    -Set ATMM debug level in LCC process for trunk <Card EIA>.
exit <Port EIA>
    - Display all exit VCC information for trunk <Port EIA>.
entry <Port EIA>
    - Display all entry VCC information for trunk <Port EIA>.
list
    - List all configured trunk ports on the local chassis.
test [<Retry Delay>]
    - Test the entry VCC consistency of all local trunk lines.
quit
    - Leave this program.
help [<option> | all]
    -Display information pertaining to <option> or "all" options.

trunk> list
Card boston.2, list of configured trunk ports:
    Port 0, UP trunk state, remote port newyork:10.0
    Port 2, UP trunk state, remote port chicago:8.2

trunk> state boston:2.2
State information for trunk , remote port chicago:8.2
    5 exit VCCs, debug level 2, local port boston:2.2
    5 entry VCCs, UP trunk state, UP state from LC

trunk> entry boston:2.2
List of entry VCCs for trunk, remote port chicago.8.2
    5 entry VCCs, UP trunk state, local port boston.2

Conn-state VCI En-id Global-conn-id
-----
ACTIVE 4 20 22 chicago:1.255[npIP:127.1.21.41][11:11:1]
ACTIVE 5 8 10 chicago:6.0[FR:200][42:42:1]
ACTIVE 3 6 8 chicago:6.1[FF:0][40:40:1]
ACTIVE 1 5 7 chicago:4.0[UNI:100][11:11:1]
ACTIVE 2 4 6 chicago:1.255[CA:boston.3][10:10:1]

trunk> exit boston:2.2
List of exit VCCs for trunk, remote port chicago.8.2
    5 exit VCCs, UP trunk state, local port boston.2.2

Conn-state VCI Ex-id En-id Global-conn-id
-----
ACTIVE 5 12 8 boston:1.255[npIP:127.1.21.44][15:15:1]
ACTIVE 4 11 7 boston:3.0[FR:21][33:33:1]
ACTIVE 3 10 6 boston:3.1[FF:0][29:29:1]
ACTIVE 2 9 5 boston:10.0[UNI:15][10:10:1]
ACTIVE 1 8 4 boston:1.255[CA:chicago.6][13:13:1]
trunk>

```

S3720

To manually balance the load across all trunks, use the global-conn-id displayed by the **entry** or **exit** command to determine which connections should be rerouted. The global-conn-id is displayed for every connection that passes through the specified port. This field indicates the source of the

connections that enter and exit the specified trunk. Since all activities related to connections are performed on either the source or destination ports of the connection (not the intermediate ports through which the connection passes), this information is very important to the rerouting activity.

The format of the global-conn-id is:

```
<chassis>.<card #>.<port #> [<conn type>:<conn name>] [ATMM identifier]
```

where

- <chassis>.<card #>.<port #> = The chassis, card number, and port number that identify the edge port from which the connection originates.
- <conn type> = The type of connection passing through the port. The possible values are:
 - FR (frame relay)
 - FF (frame forwarding)
 - UNI (ATM UNI)
 - npIP (IP between NPs)
 - CA (congestion avoidance)

Note You cannot manually reroute npIP or CA connections. The system creates them and maintains them.

- <conn name> = The name of the particular connection. This value varies depending on the connection type as shown in Table 8-2.

Table 8-2 Connection Types and Names

Connection Type	Connection Name
Frame Relay	Source DLCI
Frame Forwarding	0 (zero) ¹
ATM UNI	Source VCI
npIP	Destination IP address
Congestion Avoidance	Internal address of the destination card.

1. There is only one connection per port so the connection does not need another identifier.

- <ATMM identifier> = This field is for internal use only. It is not used in the rerouting activity.

When a failed trunk is restored, the **entry** and **exit** commands can be used to display the sources of the VCCs passing through the trunk. Once the sources are known, the VCCs can be rerouted, balancing the load across all the available trunks. Rerouting one of the VCCs (one half of the permanent virtual circuit) reroutes the whole PVC.

Each of the VCCs belongs to one of the following PVCs that run across the trunk and provide bidirectional communication between two end points:

- Frame relay
- Frame forwarding

- ATM UNI

Note Usually, both halves of a PVC are routed over the same path. However, it is possible that the two VCCs that make up each PVC are routed over different paths. If you are familiar with your network configuration, you will know if any of your PVCs are made up of a pair of VCCs that are routed over two separate paths.

Optimizing the Load Across Trunks

This procedure explains how to take down a particular connection and bring it back up so that it is rerouted by the LightStream switch. Since the switch routes the connections by selecting the path with the most available bandwidth, this process should balance the load across all of the available trunks.

When a failed trunk is restored, the restored trunk may be underutilized because the network does not automatically shift the connections back to their original paths. Therefore, you must manually reroute the connections back to the restored trunk. The LightStream network does not allow you to actually select the path through the network that the connection will follow. You must take down a connection and bring it back up so that the LightStream network will reroute it.

Note This procedure causes a momentary loss of the connection while the connection is taken down and then rerouted.

The trunkmon program displays information about individual VCCs, rather than the pairs of VCCs that make up the PVCs. Before rerouting any connections, you should run the **entry** and **exit** commands on the specified trunk port, then match up the pair of VCCs that make up each PVC. To reroute the PVC, you reroute only one of its VCCs (if the two VCCs follow the same path). The LightStream switch automatically reroutes both VCCs.

Procedure

Step 1 Use the trunkmon program as described in the section “Using The trunkmon program” to determine which connections should be rerouted.

Step 2 Set the target switch to the LightStream switch containing the source of the connection you want to reroute by entering the following at the cli> prompt:

```
cli> set snmp hostname <host name>
```

where

<host name> = The name (a text string) or IP address of the LightStream switch to which you want to set the target.

Step 3 To manually reroute a frame forwarding connection, enter the following at the cli> prompt:

```
cli> set port <port#> frameforwarding deactivate
```

where

<port#> = The frame forwarding port at either end of the frame forwarding connection. You must specify either the source or destination port of the connection. The port number is in card.port format (card = 2 - 10; port = 0 - 7 for ports on an LS line card or 0 - 1 for ports on an MS line card).

This command takes down the connection. After it has been deactivated, enter the following to reroute the connection back through its original trunk:

```
cli> set port <port#> frameforwarding activate
```

Step 4 To restore a frame relay connection, begin by enter the following at the cli> prompt:

```
cli> set port <port#> dlci <dlci#> deactivate
```

where

- <port#> = The frame relay port at either end of the frame relay connection. You must specify either the source or destination port of the connection. The port number is in card.port format (card = 2 - 10; port = 0 - 7 for ports on an LS line card or 0 - 1 for ports on an MS line card).
- <dlci#> = The data link connection identifier (DLCI) number associated with that end of the frame relay connection. You must specify the DLCI associated with the source or destination port of the connection.

This command takes down the connection. After it has been deactivated, enter the following to restore the connection:

```
cli> set port <port#> dlci <dlci#> activate
```

Step 5 To restore an ATM UNI connection, enter the following at the cli> prompt:

```
cli> set port <port#> atm-vci <atm-vci#> deactivate
```

where

- <port#> = The ATM UNI port at either end of the ATM UNI connection. You must specify the source or destination port of the connection. The port number is in card.port format (card = 2 - 10; port = 0 - 7 for ports on an LS line card or 0 - 1 for ports on an MS line card).
- <atm-vci#> = The virtual channel identifier (VCI) number associated with that end of the ATM UNI connection. You must specify the ATM VCI number associated with the source or destination port of the connection.

This command takes down the connection. After it has been deactivated, enter the following to reroute the connection back through its original trunk:

```
cli> set port <port#> atm-vci <atm-vci#> activate
```

Step 6 Repeat Step 2 through Step 5 on each of the connections you want to reroute. You may need to reroute multiple connections that originate on a single LightStream switch or you may have to reroute connections that originate on more than one LightStream switch.

Expected Results

Whenever you deactivate a connection and then reactivate it, momentary data loss occurs while the connection is taken down and then rerouted. If you do not enter the **set port activate** command immediately after you deactivate the connection, you may increase the delay before the connection

is reestablished. You can reduce the delay by placing the commands to deactivate and then activate the connections in a CLI script file. This ensures that both commands are run and minimizes the delay. For instructions on writing CLI script files, refer to the section “Creating CLI Script Files.”

However you enter the commands, the rerouting occurs very quickly and any lost data should be retransmitted by the sending application’s retransmission facility. If you are concerned about losing critical data during the rerouting process, you should make arrangements to stop passing that data during the rerouting procedure.

Appendix A: Field Descriptions

This appendix contains an alphabetical list and description of all fields that may appear in a screen display as the result of a command line interface (CLI) command.

Table 8-3 Field Names and Definitions

Field Name	Definition
Active LMI System	The local management interface (LMI) for a frame relay port.
Address Length	The length of the address.
Admin CSU Type	The channel service unit (CSU) type for the specified port.
Admin DCE Bit Rate	The speed per second set for the DCE.
Admin DSE Bit Rate	The speed per second set for the DSE.
Admin Expected DTE Rate	The expected rate of the data terminal equipment (DTE) for the specified port.
Admin Net Interface Type	The administrative net interface type for the specified port.
Admin Protocol	The administrative protocol for the specified port.
Admin Status	The administrative status (up or down) of the named port.
Administrative Status	The administrative status (up or down) of the named card.
Application	This indicates the current condition of the application as collected by the test and control system (TCS) for a particular card in the chassis. The possible values are enabled (activated) or disabled (not activated).
Application Load	This indicates the current condition of the Application Load as collected by the test and control system (TCS) for a particular card in the chassis.
ATM Data Switch Contact	The name of the person to contact for issues relating to this switch. The information should include how to contact this person.
ATM-UNI VCI list	This displays the list of ATM-UNI VCI connections. If a connection is down, an asterisk (*) appears in the state (S) column for that connection.
Banner	The program name, version number, and date display.
Begin Time	This indicates the begin time of the specified collection. (A collection runs from its begin time to its ending time. If the begin and ending times are not specified, the collection runs continuously.)
Board Initialization	This indicates the condition of the board initialization as collected by the test and control system (TCS) for a particular card in the chassis.
Bottom Temperature	The temperature indicated by the bottom sensor of the named card.
Broadcast Support	The type of broadcast supported for the specified port.
Cable Length (in feet)	The length of the cable connected to this trunk port.
Call Setup Backoff Time	The number that determines the successively longer periods of time that the system will wait before retrying call set-ups.
Date/Time	The current date and time.
Debug	The level of the CLI debug attribute. Values are on (debugging enabled) or off (debugging disabled, the default).

Field Name	Definition
Default Router	The address of the default router, if one exists.
Description	This identifies the type of device. The information should include the full name and version identification of the device.
Dest Admin DLCI	The data link connection identifier at the destination port.
Dest Admin Node	The destination node for the specified port.
Dest Oper DLCI	The data link connection identifier (DLCI) of the LightStream port at the other end of the frame relay virtual circuit.
Dest Oper Insured Burst	The maximum amount of data (in bytes or cells) that the LightStream network will transfer under normal conditions during the measurement interval from the destination port to the source port.
Dest Oper Insured Rate	The data throughput specification (in bps or cps) that the LightStream network is committed to support under normal network conditions.
Dest Oper Max Burst	The maximum amount (insured plus uninsured) data (in bytes or cells) that the LightStream network will attempt to deliver under normal conditions from destination to source during the measurement interval.
Dest Oper Max Rate	The maximum amount (insured plus uninsured) data (in bps or cps) that the LightStream network will attempt to deliver under normal conditions from destination to source.
Dest Oper Port	The LightStream port at the other end of the frame relay, frame forwarding, or ATM UNI virtual circuit.
Dest Oper VCI	The virtual channel identifier (VCI) for this ATM UNI VCC.
Dest Operational Node	The destination node.
Dest Operational Port	The destination port.
Discarded Output Packets	Number of output packets discarded due to resource limitation. If the statistic has increased since the last polling, the increase is displayed by the rate of increase on this port.
Discarded Packets Rcvd	The number of received packets that were received but discarded on this port.
Discarded Rcvd Packets	The number of packets discarded due to resource limitation. If the statistic has increased since the last polling, the increase is displayed by the rate of increase on this port.
DS3 Line Type	The type of DS3 line used on this ATM UNI port.
Call Setup Retry Time	The waiting period between the first two attempts to establish a connection on this port. If the second attempt fails, Call Setup Backoff Time is invoked.
Card	Depending on the CLI command you entered, this indicates the name of the card in the named slot or the current condition of the specified card as collected by the test and control system (TCS).
Card Name	The name of an NP, switch, or line card.
Card PID	The process identification number (PID) of the specified card.
Card Type	This indicates the type of card in the named slot.
Cards Managed by Gid	The cards (listed by chassis name and slot number) managed by the global information distribution (GID) process.
Cards Managed by ND	The cards managed by the neighborhood discovery (ND) process.
Chassis ID	This identifies the chassis by number.
Client Announcements Received	The number of client announcements received.

Field Name	Definition
Client Announcements Transmitted	The number of client announcements transmitted.
Clients Managed by Gid	The clients (listed by process identification number) managed by the global information distribution (GID) process.
Clock	This indicates the current condition of the clock as collected by the test and control system (TCS) for a particular card in the chassis.
Collection Interval	This indicates the frequency (in seconds) that information is collected in the named collection. (The default is 60 seconds.)
Collection Items	This indicates the MIB objects being collected in the named collection.
Collection Status	This indicates the status of the named collection. Values are Under Creation (stopped), Waiting (not running), or Valid (running).
Community	The name of the SNMP community.
Config DB Active	The active configuration data base.
Configuration Author	The creator of the current configuration.
Configuration Host	The name of the host where the configuration was created.
Configuration ID	The identifying number of the current configuration.
Console Trap Level	The level set for the console traps. Values are SNMP, Oper, Info, Trace, Debug, or off.
Contact	The name of the person to contact for issues relating to this switch.
CP POST	The state of power of self test (POST) commands on the card, either enabled or disabled.
Echo source	The value of the echo source attribute. If the attribute value is <i>on</i> (default), the commands in script files are displayed as they are executed. If the value is <i>off</i> , the commands are not displayed.
Ending Time	This indicates the end time for the specified collection. (A collection runs from its begin time to its ending time. If the begin and ending times are not specified, the collection runs continuously.)
Errs	The switch error statistics for neighborhood discovery (ND) process (given for both In Cells and Out Cells).
Ethernet Address	The IP address for the NP's Ethernet interface. This address is not associated with a particular NP or slot; it points to the active NP in the chassis that is used for network management.
Ethernet IP Mask	The Ethernet IP Mask of the IP address.
File	The name of a collection record.
File Size	The size (in KB) of a collection record.
Finish Time	This indicates the begin time of the specified collection.
Flash	A read only memory (ROM) that can be erased at common signal levels.
Flash Initialization	This indicates the condition of the flash initialization as collected by the test and control system (TCS) for a particular card in the chassis.
Frame forwarding connections list	This displays the list of frame forwarding connections. If a connection is down an asterisk (*) appears in the state (S) column for that connection.
Frame Relay DLCI connections list	The list of frame relay data link connection identifiers (DLCIs).
Frame Relay DLCI list	This displays the list of frame relay DLCI connections. If a connection is down an asterisk (*) appears in the state (S) column for that connection.

Field Name	Definition
Full Enquiry Interval	The number of status enquiry intervals that pass before a full status enquiry message is issued.
GID Process ID (PID)	The process identification number (PID) of the global information distribution (GID) process.
Hostname	The name of the current target switch.
Initstring	The current contents of the modem initialization string for the switch card (stored in EEPROM on the midplane).
Interval	The time interval for collection of data collection for a collection.
IP Addresses Managed by Gid	This field identifies the Internet addresses managed by the global information distribution (GID) process.
iso	The highest level object of the MIB tree.
iso.org	The subtree below the iso level of the MIB tree.
LC Software Version	This indicates the version of the LC software.
LCC Software Version	This indicates the version of the line card control process (LCC) that is running.
Line edit	The value of the CLI line edit attribute. If the line edit attribute value is <i>on</i> (default), you have access to an Emacs-like editor. If the attribute value is <i>off</i> , line edit characters are not supported.
Link Transmit Utilization	The instantaneous measure of the number of cells per second leaving the port. It includes all types of cells, data as well as control. Therefore, the Link Transmit Utilization can be higher than the Port Data Cell Capacity.
Local LMI State	The local management interface (LMI) that is active on the frame relay port.
Location	The physical location of the device.
Logging	The indication that the logging attribute is off (default) or on. If the logging attribute value is <i>off</i> , no logging takes place. To set the logging attribute to <i>on</i> , enter a log file name as the value.
Max Query Period	The maximum length of time that unanswered status enquiries are tolerated before the system declares the LMI port unreliable at the network end.
Max Status Query Errs	The maximum number of unanswered status enquiries tolerated before the system declares the LMI port unreliable at the network end.
Max Supported VCs	The maximum number of virtual circuits (VCs) allowed for this interface.
Maximum Interval between Permit Limit Updates	The maximum interval specification (in milliseconds) for trunk and outgoing edge cards to report permit limits.
Measured Bit Rate	The received bit rate for a frame relay or frame forwarding port on a low-speed edge line card.
Memory Allocation Failures	The number of memory allocation failures.
Memory In Use	This indicates the amount of memory (given in bytes) in use by the specified process.
MIB2 Type	The MIB2 type as shown for the specified port.
Minimum Interval between CA Updates	The minimum interval specification (in milliseconds) at which congestion avoidance information processes distribute aggregated congestion avoidance (CA) updates to input edge cards.
Minimum Interval between Permit Limit Updates	The minimum interval specification (in milliseconds) for trunk and outgoing edge cards to report permit limits.
MMA Collection Size	The amount of memory available to the Master Management Agent (MMA) for data collection.

Field Name	Definition
MMA PID	The process identification number of the MMA.
MMA Trap Filter Level	The priority level of traps sent from the MMA to the CLI or an NMS. Priority levels are 1 - operational, 2 - informational, 3 - trace, and 4 - debug.
MMA Trap Logging State	The trap logging state of the MMA. The settings are on (default) or off.
Multicast Packets Rcvd	Number of broadcast/multicast packets delivered (a portion of the total). If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Multicast Packets Sent	Number of broadcast/multicast packets sent (a portion of the total). If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Name	Depending on the CLI command you entered, this field may specify a card, collection, collection item, MIB object, node, process (alias name), traplog file, or groups file.
ND Clients	The list of neighborhood discovery (ND) clients.
ND Process ID (PID)	This indicates the process identification number (PID) of the neighborhood discovery (ND) process.
ND Switch Statistics	The statistics for slot, in cells, out cells, and errors for the neighborhood discovery (ND) process.
Neighbor Generic Announcements Received	The number of Neighbor Generic Announcements received.
Neighbor Generic Announcements Transmitted	The number of Neighbor Generic Announcements transmitted.
Neighbor IP Announcements Received	The number of Neighbor IP Announcements received.
Neighbor IP Announcements Transmitted	The number of Neighbor IP Announcements transmitted.
Neighbor Link Announcements Received	The number of Neighbor Link Announcements received.
Neighbor Link Announcements Transmitted	The number of Neighbor Link Announcements transmitted.
Neighbor New Announcements Received	The number of Neighbor New Announcements received.
Neighbor New Generic Announcements Received	The number of Neighbor New Generic Announcements received.
Neighbor New Link Announcements Received	The number of Neighbor New Link Announcements received.
Neighbor NPs known to ND	The neighbor network processors (NPs) known to the neighborhood discovery (ND) process.
Neighborhood Announcements Received	The number of Neighbor Announcements received.
Neighborhood New Announcements Received	The number of Neighborhood New Announcements received.
Neighbors in Exchange Start State	The number of neighbors in exchange start state.
Neighbors in Exchange State	The number of neighbors in exchange state.
Neighbors in Existent Sync State	The number of neighbors in existent sync state.

Field Name	Definition
Neighbors in Full Sync State	The number of neighbors in full sync state.
Neighbors in Loading Sync State	The number of neighbors in loading sync state.
Neighbors Managed by Gid	The list of neighbors managed by the global information distribution (GID) process.
Net Interface Type	The type of frame relay network interface on this port. The types are user network interface (UNI) and network to network interface (NNI).
New Neighbor IP Announcements Received	The number of New Neighbor IP Announcements received.
Normal Packets Rcvd	Number of unicast packets received by the indicated port. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Normal Packets Sent	Number of unicast packets sent by the indicated port. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Number of Line Cards managed by ND	/The number of line cards managed by the neighborhood discovery (ND) process.
Octets Rcvd	Total octets received from the media from the indicated port. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Octets Sent	Total octets sent on the media by the indicated port. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Oper CSU Type	The channel service unit (CSU) type for the named port.
Oper DCE Bit Rate	The DCE bit rate for the named port.
Oper Expected DTE Rate	The expected DTE rate for the named port.
Oper Interval	The channel service unit (CSU) type for the named port.
Oper Net Interface Type	The interface type for the named port.
Oper Protocol	The protocol operating on the interface.
Oper Status	The operational status (up or down) of the named port.
Operational Max Frame Size	The maximum frame size (in bytes) for the named port.
Operational Status	This indicates the status (up or down) of the card in the named slot.
Output Errors	Packets discarded due to error. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Paddle Card	This indicates the current status of the access card (presence and condition) as collected by the test and control system (TCS) for a particular card in the chassis.
Paddle Card Override	This indicates the current condition of the access card override (enabled or disabled) as collected by the test and control system (TCS) for a particular card in the chassis.
Paddle Power Override	This indicates the current condition of the access power override (enabled or disabled) as collected by the test and control system (TCS) for a particular card in the chassis.
PID	The process identification number.
PID Administrative Status	This indicates the administrative status (active or inactive) of the named process.
PID Alias	This indicates the alias name for the specified process.
PID Name	This indicates the name for the specified process.
PID Operation Status	The operation status of the process.

Field Name	Definition
PID Trap Level	This indicates the trap level setting for the specified process. Values are oper, info (default), trace, or debug.
PID Up Time	The length of time the process has been running.
CP Scrambling	This indicates whether the cell payload scrambling is enabled or disabled for the named card. (The default is disabled.)
Port Data Cell Capacity	The available data cell capacity for the named port.
Port Frame Forwarding Name	The name of the frame forwarding port.
Port Frame Relay Name	The name of the frame relay port.
Port MTU	The maximum transmission unit number for the specified port.
Port Name	The name of the specified port.
Port Speed	The speed (in bps) for the specified port.
Port Type	The port type (MS Trunk or LS Edge, for example).
Port Unreserved Capacity	The available capacity (in cells) for the named port.
Ports Managed by Gid	The list of ports managed by the global information distribution process.
POST	This indicates the current condition of the power on self test (POST) as collected by the test and control system (TCS) for a particular card in the chassis.
Power Supply	This indicates the current condition of the power supply as collected by the test and control system (TCS) for a particular card in the chassis.
Power Supply A	This indicates the condition of the bulk power tray in power slot A of the LightStream chassis. If this slot is unused, Empty will appear in this field.
Power Supply A Type	This indicates the power type in power slot A of the LightStream chassis. If this slot is unused, Empty will appear in this field.
Power Supply B	This indicates the condition of the bulk power tray in power slot B of the LightStream chassis. If this slot is unused, Empty will appear in this field.
Power Supply B Type	This indicates the power type in power slot B of the LightStream chassis. If this slot is unused, Empty will appear in this field.
Primary Addr	This indicates the IP address for the primary NP's switch interface.
Primary Switch	This identifies the primary active switch card (SA or SB).
PROGRAM:	The program name and its compile time.
Receive Errors	The packets discarded due to format error. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
Registered ND Client Processes	The list of registered neighborhood discovery (ND) processes.
Remote LMI State	The state of the remote LMI. LMI is local management interface; a frame relay protocol for getting the status of frame relay circuits from attached frame relay devices.
Request Interval	The maximum number of seconds specified that the system expects to elapse between status enquiry messages from the user end of the frame relay connection.
SCSI Power	This indicates the current condition of the SCSI power as collected by the test and control system (TCS) for a particular card in the chassis.
Secondary Addr	This indicates the IP address for the secondary NP.
Slot #	This indicates the type of card in Slot #. If the slot is unused, Empty will appear in this field.
Slot # Config Assembly	This identifies the configuration assembly as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.

Field Name	Definition
Slot # Config Postcode	This identifies the configuration post code as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Config Serialnum	This identifies the configuration serial number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Config Slavecode	This identifies the configuration slave code as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Config Type	This identifies the configuration type as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Daughter Assembly	This identifies the daughter assembly number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Daughter Serialnum	This identifies the daughter serial number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Oem Assembly	This identifies the OEM assembly number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Oem Serialnum	This identifies the OEM serial number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Paddle Assembly	This identifies the access card assembly number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Paddle Serialnum	This identifies the paddle serial number as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # State	The temperature conditions (top and bottom) as collected by the test and control system (TCS) for the card in the indicated slot of the chassis.
Slot # Voltage <ul style="list-style-type: none"> • TCS VCC Voltage • VCC Voltage • SCSI Voltage • VPP Voltage 	This indicates the TCS VCC, VCC, SCSI, and VPP voltages as collected by the test and control system (TCS) for the card in the indicated slot of the chassis. The current voltages are shown as well as the normal voltage ranges.
Slot 2	This indicates the type of card in slot 2. If the slot is unused, Empty will appear in this field.
Slot 3	This indicates the type of card in slot 3. If the slot is unused, Empty will appear in this field.
Slot 4	This indicates the type of card in slot 4. If the slot is unused, Empty will appear in this field.
Slot 5	This indicates the type of card in slot 5. If the slot is unused, Empty will appear in this field.
Slot 6	This indicates the type of card in slot 6. If the slot is unused, Empty will appear in this field.
Slot 7	This indicates the type of card in slot 7. If the slot is unused, Empty will appear in this field.
Slot 8	This indicates the type of card in slot 8. If the slot is unused, Empty will appear in this field.
Slot 9	This indicates the type of card in slot 9. If the slot is unused, Empty will appear in this field.
Slot 10	This indicates the type of card in slot 10. If the slot is unused, Empty will appear in this field.
Slot of Primary NP	This indicates the slot that contains the primary active NP.

Field Name	Definition
Slot of This NP	This indicates the slot that contains the NP being displayed.
Slot SA	This indicates the type of card in slot SA. If the slot is unused, Empty will appear in this field.
Slot SB	This indicates the type of card in slot SB. If the slot is unused, Empty will appear in this field.
Software Version Number	The version number of the specified application.
Source Node	The node at the source of the service.
Source Port	The port at the source of the service.
Source VCI	The source virtual channel identifier (VCI) for the VCC.
Src Admin Insured Burst	The maximum amount of data (in bytes or cells) that the LightStream network will transfer under normal conditions from the destination port to the source port during the measurement interval.
Src Admin Insured Rate	The data throughput (in bits per second or cells) that the LightStream network is committed to support under normal network conditions. The insured rate (IR) is specified in bits per second for FF and FR interfaces, in cells per second for ATM UNI.
Src Admin Max Burst	The maximum amount (insured plus uninsured) data (in bytes or cells) that the LightStream network will attempt to deliver under normal conditions from destination to source during the measurement interval. (Measurement interval is calculated by dividing maximum burst size by maximum rate.)
Src Admin Max Rate	The maximum amount (insured plus uninsured) data (in bps or cps) that the LightStream network will attempt to deliver under normal conditions from destination to source. (The uninsured data may be dropped if the network is congested.)
Src DLCI	The LightStream node at the other end of the frame relay, frame forwarding, or ATM UNI virtual circuit.
Src Node	The node at the source of the service.
Src Oper Insured Burst	The data throughput on a given virtual circuit that the LightStream network commits to transfer during a specified interval.
Src Oper Insured Rate	The data throughput specification in bps or cps that the LightStream network supports under normal network conditions.
Src Oper Max Burst	The maximum insured plus uninsured data throughput on a given virtual circuit that the LightStream network commits to transfer during a specified interval, in bytes (for FF and FR) or cells (for ATM UNI),
Src Oper Max Rate	The maximum, insured plus uninsured data throughput rate that the LightStream network will attempt to deliver on a given virtual circuit. The uninsured data may be dropped if the network is congested. This throughput is the highest that the virtual circuit will ever deliver.
Src Port	The LightStream port at the other end of the frame relay, frame forwarding, or ATM UNI virtual circuit.
State	The state of the cards managed by the neighborhood discovery (ND) process.
Status Query Period	The maximum length of time that unanswered status enquiries are tolerated before the system declares the LMI port unreliable at the network end.
Subnet Mask	The address mask used to identify which bits in the address are network significant, subnet significant, and host significant portions of the complete address.

Field Name	Definition
System Up Time	The length of time the system has been up. The time is given in hours, minutes, and seconds.
TCS Hub	This indicates the current condition of the TCS hub as collected by the test and control system (TCS) for a particular card in the chassis.
TCS VCC Power	This indicates the current condition of the test and control system (TCS) VCC (+5V) power as collected by the TCS for a particular card in the chassis.
TCS Voltage	This indicates the TCS voltage of the named card.
Temperature	This indicates the current temperature condition collected by the test and control system (TCS) for a particular card in the chassis.
Temperature Bottom	The temperature indicated by the bottom sensor of the named card.
Temperature Paddle Card Region 1	The temperature in region 1 of the named access card.
Temperature Paddle Card Region 2	The temperature in region 2 of the named access card.
Temperature Top	The temperature indicated by the bottom sensor of the named card.
Terminal Type	This indicates the terminal type you are using.
Timer	The CLI timer that indicates the time since the CLI was restarted or since this timer was reset.
Top Temperature	The temperature indicated by the top sensor of the named card.
Traplevel	The level that the CLI debug attribute has been set to. (Values are off, oper, info, trace, or debug.)
Type	The type of service for this ATM UNI circuit (guaranteed or insured).
Unknown Protocols Rcvd	The number of packets received that were destined for unknown protocols. If the statistic has increased since the last polling, the increase is displayed by the rate of increase.
User Monitored Events	The number of monitored events.
User Polling Interval	The number of seconds specified between consecutive status enquiries sent by the user portion of a frame relay interface that has a local management interface (LMI).
Value	The alias name of a PID.
VCC Power	This indicates the current condition of the VCC (+5 V) power as collected by the TCS for a particular card in the chassis.
VCC Voltage	This indicates the VCC (+5 V) voltage of the named card.
VEE Power	This indicates the current condition of the VEE power as collected by the TCS for a particular card in the chassis.
VPP Power	This indicates the current condition of the VPP power as collected by the TCS for a particular card in the chassis.
XILINX Load	This indicates the current condition of the XILIN Load as collected by the test and control system (TCS) for a particular card in the chassis.