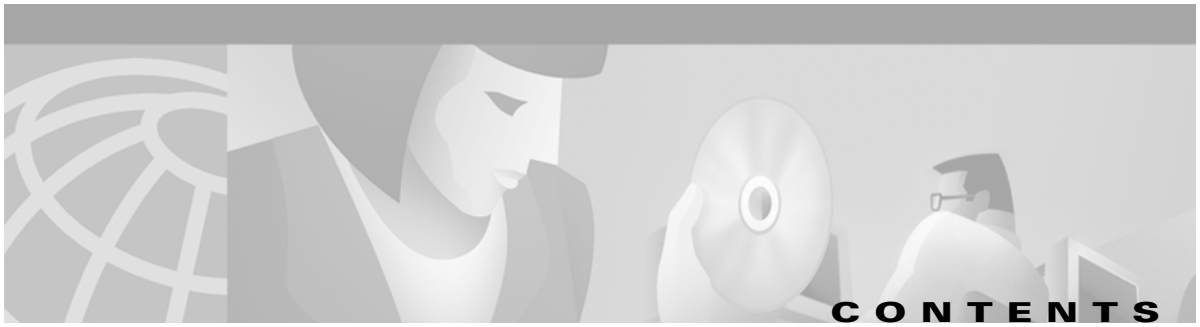
Configuration Guide for Cisco DSLAMs with NI-2

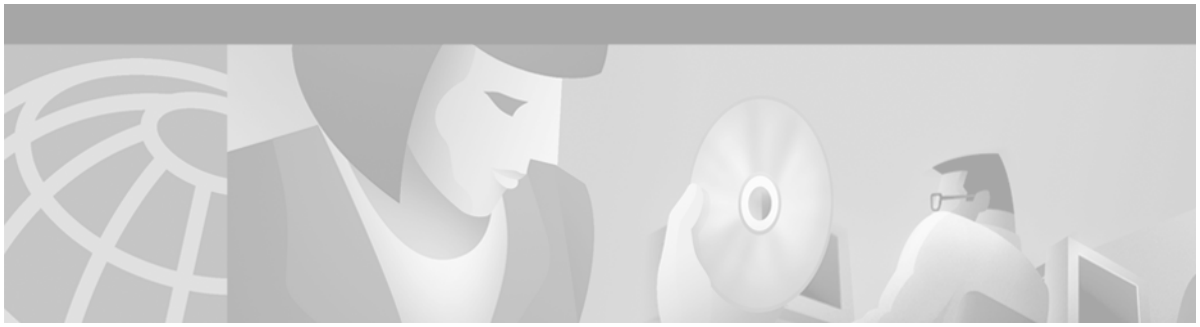
Cisco IOS Release 12.1DA
July 23, 2001

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number:
Text Part Number: 78-6691-04



CONTENTS



About This Guide

This preface tells you who should read the Configuration Guide for Cisco DSLAMs with NI-2, how the document is organized, and the document conventions it follows.

Audience

This guide is written for anyone who installs or operates Cisco digital subscriber line access multiplexers (DSLAMs) with NI-2. This includes the:

- Cisco 6015
- Cisco 6130
- Cisco 6160
- Cisco 6260

How This Guide Is Organized

This guide includes 17 chapters and an index. They are:

Chapter	Title	Content
Chapter 1	Cisco DSLAM User Interface	Describes the DSLAM user interface and provides instructions for using the command-line interface. Describes how to access and list the commands available in each command mode, and explains the primary uses for each command mode.
Chapter 2	Configuring Terminal Lines and Modem Support	Explains how to configure lines, modems, and terminal settings to access the ATM switch for management purposes.
Chapter 3	Initially Configuring the Cisco DSLAM	Describes the initial configuration of the Cisco DSLAM.
Chapter 7	Configuring Digital Subscriber Lines	Describes how to configure the DSLAM for Digital Subscriber Line (DSL) service.

Chapter 13	Configuring In-Band Management	Describes how to configure in-band management for the DSLAM.
Chapter 16	Configuring the Trunk and Subtended Interfaces	Describes the steps required to configure the trunk and subtended interfaces on the DSLAM NI-2 card.
Chapter 17	Loading System Software Images and Configuration Files	Describes how to load and maintain system software images and configuration files.
Index		

Other information necessary for ATM configuration tasks available on DSLAMs is contained in the *ATM Switch Router Software Configuration Guide*.Table 1provides manual titles and links to detailed documentation and configuration examples for the ATM configuration tasks available on DSLAMs.

Table 1 ATM Configuration Tasks

ATM Configuration Tasks	ATM Switch Router Software Configuration Guide	Hyperlink
Configuring System Management Functions	Chapter 4	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/sysadmin.htm
Configuring ATM Network Interfaces	Chapter 5	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/if_config.htm
Configuring Virtual Connections	Chapter 6	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/vir_circ.htm
Configuring Operation, Administration, and Maintenance	Chapter 7	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/op_maint.htm
Configuring Resource Management	Chapter 8	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/rm_cnf.htm
Configuring ILMI	Chapter 9	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/ilmi_cnf.htm
Configuring ATM Routing and PNNI	Chapter 10	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/pnni_cnf.htm
Using Access Control	Chapter 11	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/access.htm

Table 1 ATM Configuration Tasks

ATM Configuration Tasks	ATM Switch Router Software Configuration Guide	Hyperlink
Configuring ATM Accounting and ATM RMON	Chapter 14	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/act_rmon.htm
Configuring Signalling Features	Chapter 16	http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/signal.htm

Conventions

This publication uses the document conventions listed in this section.

Table 2 Font Conventions

Convention	Definition	Sample
Times bold	Text body font used for any argument, command, keyword, or punctuation that is part of a command that the user enters in text and command environments. Also used for names of some GUI elements.	This is similar to the UNIX **route** command.
Times italic	Text body font used for publication names and for emphasis.	See the *Cisco 6100 Series User Guide* for further details.
`Courier`	Font used for screen displays, prompts, and scripts.	`Are you ready to continue? [Y]`
`Courier bold`	Font used to indicate what the user enters in examples of command environments.	Login: **`root`** Password: **`<password>`**

Table 3 Command Syntax Conventions

Convention	Definition	Sample
Vertical bar (|)	Separates alternative, mutually exclusive elements.	**offset-list** {**in** | **out**} *offset*
Square brackets ([ ])	Indicate optional elements.	[**no**] **offset-list** {**in** | **out**} *offset*
Braces ({ })	Indicate a required choice.	**offset-list** {**in** | **out**} *offset*
Braces within square brackets ([{ }])	Indicate a required choice within an optional element.	[{*letter*|*number*}**Enter**]

Table 3 Command Syntax Conventions (continued)

Convention	Definition	Sample
Boldface	Indicates commands and keywords that are entered literally as shown	[**no**] **offset-list** {**in** | **out**} *offset*
Italics	Indicate arguments for which you supply values. **Note** In contexts that do not allow italics, arguments are enclosed in angle brackets (< >).	**offset-list** {**in** | **out**} *offset*

Note fMeans *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information or information that might save time.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warning **Means *danger*. You are in a situation that could cause bodily injury. Before you work on any equipment, you must be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. To see translated versions of the warning, refer to the *Regulatory Compliance and Safety* document that accompanied the device.**

Related Documentation

Hardware Documents

A complete list of all DSL hardware product related documentation is available on the World Wide Web at http://www.cisco.com/univercd/cc/td/doc/product/dsl_prod/index.htm.

Software Documents

A complete list of all DSL IOS software product related documentation is available on the World Wide Web at http://www.cisco.com/univercd/cc/td/doc/product/dsl_prod/ios_dsl/index.htm.

In the ATM software product related documentation, look for information pertaining to the Cisco LightStream 1010, which uses the same software base as the NI-2 DSL systems. This documentation is available on the World Wide Web at http://www.cisco.com/univercd/cc/td/doc/product/atm/index.htm.

Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at the following sites:

- http://www.cisco.com
- http://www-china.cisco.com
- http://www-europe.cisco.com

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or as an annual subscription.

Ordering Documentation

Cisco documentation is available in the following ways:

- Registered Cisco Direct Customers can order Cisco Product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click Feedback in the toolbar and select Documentation. After you complete the form, click Submit to send it to Cisco.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Attn Document Resource Connection
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools. For Cisco.com registered users, additional troubleshooting tools are available from the TAC website.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

Cisco.com provides a broad range of features and services to help customers and partners streamline business processes and improve productivity. Through Cisco.com, you can find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online technical support, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on Cisco.com to obtain additional personalized information and services. Registered users can order products, check on the status of an order, access technical support, and view benefits specific to their relationships with Cisco.

To access Cisco.com, go to the following website:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC website is available to all customers who need technical assistance with a Cisco product or technology that is under warranty or covered by a maintenance contract.

Contacting TAC by Using the Cisco TAC Website

If you have a priority level 3 (P3) or priority level 4 (P4) problem, contact TAC by going to the TAC website:

http://www.cisco.com/tac

P3 and P4 level problems are defined as follows:

- P3—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- P4—You need information or assistance on Cisco product capabilities, product installation, or basic product configuration.

In each of the above cases, use the Cisco TAC website to quickly find answers to your questions.

To register for Cisco.com, go to the following website:

http://www.cisco.com/register/

If you cannot resolve your technical issue by using the TAC online resources, Cisco.com registered users can open a case online by using the TAC Case Open tool at the following website:

http://www.cisco.com/tac/caseopen

Contacting TAC by Telephone

If you have a priority level 1 (P1) or priority level 2 (P2) problem, contact TAC by telephone and immediately open a case. To obtain a directory of toll-free numbers for your country, go to the following website:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

P1 and P2 level problems are defined as follows:

- P1—Your production network is down, causing a critical impact to business operations if service is not restored quickly. No workaround is available.
- P2—Your production network is severely degraded, affecting significant aspects of your business operations. No workaround is available.

CHAPTER 1

Cisco DSLAM User Interface

This chapter describes the Cisco DSLAM user interface, provides instructions for using the command-line interface, describes how to use the help system and also describes the command editing and command history features that enable you to recall previous command entries and edit previously entered commands.

This chapter includes the following sections:

Understanding the User Interface

The Cisco DSLAM user interface provides access to several different command modes, each with related commands. For security, the user interface provides three levels of access to commands:

- User mode—Called user EXEC mode
- Privileged mode— The privileged mode is called privileged EXEC mode and requires a password. The unprivileged user mode is called user EXEC mode.

  Note Because all commands available in user EXEC mode are also available in privileged EXEC mode, user EXEC mode is referred to as EXEC mode in this guide.

  From the privileged EXEC mode, you can access global configuration mode and three specific configuration modes:

  - Terminal
  - Memory
  - Network configuration

- Read-only memory (ROM) monitor mode—This mode accesses a basic system kernel to which the DSLAM may default at startup if it does not find a valid system image, or if its configuration file is corrupted.

You can enter commands in uppercase, lowercase, or both. Only passwords are case sensitive. You can abbreviate commands and keywords to a unique number of characters. For example, you can abbreviate the **show** command to **sh**. After you enter the command line at the system prompt, press **Return** to execute the command.

Most configuration commands have a **no** form. In general:

- Use the **no** form of a command to disable a feature or function
- Use the command without the **no** keyword to reenable a disabled feature or enable a feature disabled by default

The context-sensitive help system allows you to obtain a list of commands available for each command mode or a list of available options for a specific command by entering a question mark (**?**).

Accessing Command Modes

This section describes how to access the DSLAM command modes. Table 1-1 lists

- The command mode names
- The method to access that mode
- The prompt you see while in that mode (For the purpose of this guide, the prompts use the default node name “DSLAM”.)
- The method to exit that mode

Note Table 1-1 does not include all of the possible ways to access or exit each command mode.

Table 1-1 Command Modes

Command Mode	Access Method	Prompt	Exit Method
EXEC (user)	Log in to the switch or DSLAM.	`DSLAM>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** command and enter your password.	`DSLAM#`	To return to user EXEC mode, use the **disable** command.
ROM monitor	From privileged EXEC mode, use the **reload** command. Press **Break** during the first 60 seconds while the system boots.	`>`	To exit to user EXEC mode, use the **continue** command.
Global configuration	From privileged EXEC mode, use the **configure** command. Use the keyword **terminal** to enter commands from your terminal.	`DSLAM(config)#`	To exit to privileged EXEC mode, use the **exit** or **end** command or press **Ctrl-Z**.

Table 1-1 Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method
Interface configuration	From global configuration mode, enter by specifying an interface with the **interface** command.	`DSLAM(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
Profile configuration	From interface configuration, enter by specifying a profile with a **dsl profile** command.	`DSLAM(cfg-dsl-profile)#`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
Line configuration	From interface configuration, enter by specifying a profile with a **line** command.	`DSLAM(config-line)#`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
ATM router configuration	From global configuration mode, configure the ATM router configuration with the **atm router pnni** command.	`DSLAM(config-atm-router)#`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
PNNI node configuration	From ATM router configuration mode, configure the PNNI routing node with the **node** command.	`DSLAM(config-pnni-node)#`	To exit to ATM router configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
ATM accounting file	From global configuration mode, define an ATM accounting file with the **atm accounting file** command.	`DSLAM(config-acct-file)#`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
ATM accounting selection	From global configuration mode, define an ATM accounting selection table entry with the **atm accounting selection** command.	`DSLAM(config-acct-sel)#`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.

Table 1-1 Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method
ATM E.164 translation table configuration	From global configuration mode, enter the **atm e164 translation-table** command	`DSLAM(config-atm-e164)`	To exit to priviledged EXEC mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.
ATM signaling diagnostics configuration	From global configuration mode, enter the command **atm signalling diagnostics** and an index to configure.	`DSLAM(cfg-atmsig-diag)`	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, use the **end** command or press **Ctrl-Z**.

Understanding Command Modes

The following section describes the various command modes and their levels of user access.

User EXEC Mode

When you log in to the DSLAM, you are in user EXEC, or simply EXEC, command mode. The EXEC mode commands available at the user level are a subset of those available at the privileged level. In general, the user EXEC mode commands allow you to connect to remote switches, change terminal settings on a temporary basis, perform basic tests, and list system information.

The user EXEC mode prompt consists of the DSLAM's host name followed by the angle bracket (>):

```
Frodo>
```

or

```
DSLAM>
```

The default host name is "DSLAM", unless it has been changed using the **host name** global configuration command.

Privileged EXEC Mode

The privileged EXEC mode command set includes all user EXEC mode commands and the **configure** command, through which you can access global configuration mode and the remaining configuration submodes. Privileged EXEC mode also includes high-level testing commands, such as **debug**, and commands that display potentially secure information.

To enter or exit priviledged EXEC mode, follow these steps:

Step	Command	Task
1.	DSLAM>**enable** Password:*password*	Enter privileged EXEC mode from EXEC mode.[1]
2.	DSLAM#	Enter priviledged EXEC commands.
3.	DSLAM#**disable** DSLAM>	Exit priviledged EXEC mode and return to EXEC mode.[2]

1. The prompt changes to the DSLAM's host name followed by the pound sign (#).
2. The prompt changes back to the DSLAM's host name followed by the angle bracket (>).

The system administrator uses the **enable password global configuration** command to set the password, which is case-sensitive. If an enable password was not set, you can access privileged EXEC mode only from the console.

ROM Monitor Mode

ROM monitor mode provides access to a basic system kernel, from which you can boot the DSLAM or perform diagnostic tests. The system may enter ROM mode automatically if the DSLAM does not find a valid system image, or if the configuration file is corrupted. The ROM monitor prompt is the angle bracket (>) without the DSLAM host name.

You can also enter ROM monitor mode by intentionally interrupting the boot sequence with the **Break** key during loading.

To return to EXEC mode from ROM monitor mode, use the **continue** command:

```
DSLAM>continue
DSLAM>
```

Global Configuration Mode

Global configuration mode provides access to commands that apply to the entire system. From global configuration mode you can also enter the other configuration modes described in these sections.

To enter global configuration mode from privileged EXEC mode, enter the **configure** command and specify the source of the configuration commands at the prompt. The prompt changes to the DSLAM's host name followed by `(config)#`:

```
DSLAM#configure
Configuring from terminal, memory, or network [terminal]? <CR>
Enter configuration commands, one per line. End with CNTL/Z.
DSLAM(config)#
```

You can specify either the terminal, nonvolatile read-only memory (NVRAM), or a file stored on a network server as the source of configuration commands. The default is to enter commands from the terminal console.

As a shortcut for accessing the terminal method of configuration, enter:

```
DSLAM#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
DSLAM(config)#
```

To exit global configuration command mode and return to privileged EXEC mode, use the **exit** or **end** command, or press **Ctrl-Z**:

```
DSLAM(config)#end
DSLAM#
```

Interface Configuration Mode

Interface configuration mode provides access to commands that apply on an interface basis. Use these commands to modify the operation of an interface such as an ATM, Ethernet, or asynchronous port.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**interface** *interface-type interface-number*	Enter interface configuration mode from global configuration mode.
3.	DSLAM(config-if)#**exit**	Exit interface configuration mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

Profile Mode

Profile mode provides access to DSL profile commands. (See "Profile Mode" section on page 1-6.)

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify a profile.
3.	DSLAM(cfg-dsl-profile)#**no alarms**	Disable alarms for that profile.
4.	DSLAM(cfg-dsl-profile)#**exit**	Exit profile mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

Line Configuration Mode

Line configuration mode provides access to commands used to configure lines on the DSLAM.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**line** *line-index*	Specify a line.
3.	DSLAM(config-line)#**exit**	Exit profile mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

ATM Router Configuration Mode

ATM router configuration mode provides access to commands used to configure Private Network-to-Network Interface (PNNI) routing.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**atm router pnni**	Enter ATM router configuration mode from global configuration mode.[1]
3.	DSLAM(config-atm-router)#**exit**	Exit ATM router configuration mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

1. The prompt changes to (config-atm-router)#.

PNNI Node Configuration Mode

The PNNI node configuration mode is a submode of ATM router configuration mode and provides access to commands you use to configure PNNI nodes on the DSLAM.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**atm router pnni**	Enter ATM router configuration mode from global configuration mode.[1]
3.	DSLAM(config-atm-router)#**node** *node-index*	Enter PNNI node configuration mode from global configuration mode.[2]
4.	DSLAM(config-pnni-node)#**exit**	Exit PNNI node configuration mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

1. The prompt changes to (config-atm-router)#.
2. The prompt changes to (config-pnni-node)#.

ATM Accounting File Configuration Mode

ATM accounting file configuration mode provides access to commands used to configure a file for accounting and billing of virtual circuits (VCs).

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.

Step	Command	Task
2.	DSLAM(config)#**atm accounting file** *accounting-filename*	Enter ATM accounting file configuration mode from global configuration mode.[1]
3.	DSLAM(config-acct-file)#**exit**	Exit ATM accounting file configuration mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

1. The prompt changes to (config-acct-file)#.

ATM Accounting Selection Configuration Mode

ATM accounting selection configuration mode provides access to commands used to specify the connection data to be gathered from the DSLAM.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**atm accounting selection** *accounting-selection-index*	Enter ATM accounting selection configuration mode from global configuration mode.[1]
3.	DSLAM(config-acct-sel)#**exit**	Exit ATM accounting selection configuration mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

1. The prompt changes to (config-acct-sel)#.

ATM E.164 Translation Table Configuration Mode

ATM E.164 translation table configuration mode provides access to commands that you use to configure the translation table that maps native E.164 format addresses to ATM end system (AESA) format addresses.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**atm e164 translation-table**	Enter ATM E.164 translation table configuration mode from global configuration mode.[1]
3.	DSLAM(config-atm-e164)#**exit** or DSLAM(config-atm-e164)#**end**	Exit ATM E.164 translation table configuration mode and return to priviledged EXEC mode.

1. The prompt changes to (config-atm-e164)#.

ATM Signaling Diagnostics Configuration Mode

ATM signaling diagnostics configuration mode provides access to commands used to configure the signaling diagnostics table.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**atm signalling diagnostics**	Enter ATM signaling diagnostics configuration mode.
3.	DSLAM(cfg-atmsig-diag)#**exit**	Exit ATM signaling diagnostics configuration mode and return to global configuration mode. Enter **end** to return to priviledged EXEC mode.

Using Context-Sensitive Help

The user interface provides context-sensitive help in all modes. This section describes how to configure and display context-sensitive help.

Configuring Help for Terminal Sessions

The following commands configure full help.

Command	Task
DSLAM#**terminal full-help**	In privileged EXEC mode, configure the current terminal session to receive help for the full set of user-level commands.
DSLAM(config-line)#**full-help**	In line configuration mode, configure a specific line to allow users without privileged access to obtain full help.

Displaying Context-Sensitive Help

To get help specific to a command mode, a command, a keyword, or argument, perform one of these tasks:

Command	Task
help	Obtain a brief description of the help system in any command mode.
*abbreviated-command-entry***?**	Obtain a list of commands that begin with a particular character string.
*abbreviated-command-entry***<Tab>**	Complete a partial command name.

Command	Task
`?`	List all commands available for a particular command mode.
command `?`	List a command's associated keywords.
command keyword `?`	List a keyword's associated arguments.

Using Word Help

To view a list of commands that begin with a particular character sequence, type those characters followed immediately by the question mark (?). Do not include a space. This form of help is called word help, because it completes a word for you.

In this example, the system displays the possible commands in privileged EXEC mode that begin with "co."

```
DSLAM#co?
configure  connect  copy
```

This form helps you determine the minimum subset that can be used when you abbreviate a command.

Command Syntax Help

To list keywords or arguments, enter a question mark (?) in place of a keyword or argument. Include a space before the ?. This form of help is called command syntax help, because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you have already entered.

This example demonstrates the use of command syntax help to complete the **access-list** command. Entering the question mark (**?**) displays the allowed arguments:

```
DSLAM(config)#access-list ?
  <1-99>     IP standard access list
  <100-199>  IP extended access list
```

Enter the access list number, **99**, followed by question mark (**?**) to display the allowed keywords:

```
DSLAM(config)#access-list 99 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
```

Enter the **deny** argument followed by a question mark (**?**) to display the next argument (host name or IP address) and two keywords:

```
DSLAM(config)#access-list 99 deny ?
  Hostname or A.B.C.D  Address to match
  any                  Any source host
  host                 A single host address
```

Enter the IP address followed by a question mark (**?**) to display a final (optional) argument. The <cr> indicates that you can press **Return** to execute the command:

```
DSLAM(config)#access-list 99 deny 131.108.134.0 ?
  A.B.C.D  Wildcard bits
  <cr>
DSLAM(config)#<cr>
```

The system adds an entry to access list 99 that denies access to all hosts on subnet 131.108.134.0.

Checking Command Syntax

The user interface provides an error indicator (^) that appears in the command string in which you have entered an incorrect or incomplete command, keyword, or argument.

This example shows a command entry that is correct up to the last element:

```
DSLAM#clock set 13:04:30 28 apr 98
                                 ^
% Invalid input detected at '^' marker.
```

The caret symbol (^) and help response indicate the location in which the error occurs. To list the correct syntax, reenter the command, substituting a question mark (**?**) where the error occurred:

```
DSLAM#clock set 13:32:00 23 February ?
  <1993-2035> Year
DSLAM# clock set 13:32:00 23 February
```

Enter the year using the correct syntax and press **Return** to execute the command:

```
DSLAM#clock set 13:32:00 23 February 1993
```

Using the Command History Features

The user interface provides a history or record of commands you enter. You can use the command history feature for recalling long or complex commands or entries, including access lists. With the command history feature, you can complete the tasks in these sections:

- Set the Command History Buffer Size
- Recall Commands
- Disable the Command History Feature

Setting the Command History Buffer Size

By default, the system records ten command lines in its history buffer. Use the following commands to set the number of command lines the system records.

Command	Task
DSLAM#**terminal history** [**size** *number-of-lines*]	In privileged EXEC mode, enable the command history feature for the current terminal session.
DSLAM(config-line)**history** [**size** *number-of-lines*]	In line configuration mode, enable the command history feature for a specific line.

Recalling Commands

To recall commands from the history buffer, perform one of these tasks:

Key Sequence/Command	Task
Press **Ctrl-P** or the up arrow key.[1]	Recall commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Press **Ctrl-N** or the down arrow key.[1]	Return to more recent commands in the history buffer after recalling commands with Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.
`DSLAM>show history`	While in EXEC mode, list the last several commands you have just entered.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Disabling the Command History Feature

The command history feature is automatically enabled. Use the following commands to disable it.

Command	Task
`DSLAM>terminal no history`	In EXEC mode, disable the command history feature for the current terminal session.
`DSLAM(config-line)#no history`	In line configuration mode, configure the line to disable the command history feature.

Using the Editing Features

The user interface includes an enhanced editing mode that provides a set of editing key functions similar to those of the Emacs editor.

Using the editing features you can perform the tasks described in these sections:

- Enable Enhanced Editing Mode
- Move Around on the Command Line
- Complete a Partial Command Name
- Paste in Buffer Entries
- Edit Command Lines that Wrap
- Delete Entries
- Scroll Down a Line or a Screen
- Redisplay the Current Command Line
- Transpose Mistyped Characters
- Control Capitalization
- Designate a Keystroke as a Command Entry

- Disable Enhanced Editing Mode

Enabling Enhanced Editing Mode

Although the current software release enables the enhanced editing mode by default, you can disable it and revert to the editing mode of previous software releases. Use the following commands to reenable the enhanced editing mode.

Command	Task
`DSLAM>terminal editing`	In EXEC mode, enable the enhanced editing features for the current terminal session.
`DSLAM(config-line)#editing`	In line configuration mode, enable the enhanced editing features for a specific line.

Moving Around on the Command Line

Use these keystrokes to move the cursor around on the command line for corrections or changes:

Keystrokes	Task
Press **Ctrl-B** or press the left arrow key.[1]	Move the cursor back one character.
Press **Ctrl-F** or press the right arrow key.[1]	Move the cursor forward one character.
Press **Ctrl-A**.	Move the cursor to the beginning of the command line.
Press **Ctrl-E**.	Move the cursor to the end of the command line.
Press **Esc B**.	Move the cursor back one word.
Press **Esc F**.	Move the cursor forward one word.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

Completing a Partial Command Name

If you cannot remember a complete command name, you can use **Tab** to allow the system to complete a partial entry:

Keystrokes	Task
Enter the first few letters and press **Tab**.	Complete a command name.

If your keyboard does not have **Tab**, press **Ctrl-I** instead.

In this example, when you enter the letters **conf** and press **Tab**, the system provides the complete command:

```
DSLAM#conf<Tab>
DSLAM#configure
```

If you enter an ambiguous set of characters, the system generates an error message. To display the list of legal commands beginning with the specified string, enter a question mark (**?**) after you see the error message. See the section "Using Word Help" section on page 1-10.

Pasting in Buffer Entries

The system provides a buffer that contains the last ten items you deleted. You can recall these items and paste them in the command line by using these keystrokes:

Keystrokes	Task
Press **Ctrl-Y**.	Recall the most recent entry in the buffer.
Press **Esc Y**.	Recall the next buffer entry.

The buffer contains only the last ten items you have deleted or cut. If you press **Esc Y** more than 10 times, you cycle back to the first buffer entry.

Editing Command Lines that Wrap

The new editing command set provides a wraparound feature for commands that extend beyond a single line on the screen. When the cursor reaches the right margin, the command line shifts 10 spaces to the left. You cannot see the first 10 characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, use these keystrokes:

Keystrokes	Task
Press **Ctrl-B** or the left arrow key[1] repeatedly	Scroll back one character at a time to the beginning of a command line to verify that you entered a lengthy command correctly.
Press **Ctrl-A**	Return directly to the beginning of the line.

1. The arrow keys function only on ANSI-compatible terminals such as VT100s.

In this example, the **access-list** command entry extends beyond one line. When the cursor reaches the end of the line, the line is shifted ten spaces to the left and redisplayed. The dollar sign ($) indicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted ten spaces to the left.

```
DSLAM(config)#access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1
DSLAM(config)#$ 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.25
DSLAM(config)#$t tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq
DSLAM(config)#$108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq 45
```

When you complete the entry, press **Ctrl-A** to check the complete syntax before pressing **Return** to execute the command. The dollar sign ($) appears at the end of the line to indicate that the line has scrolled to the right:

```
DSLAM(config)#access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1$
```

The DSLAM default is a terminal screen that is 80 columns wide. If you have a width other than that, use the **terminal width** command to provide correct width of your terminal.

Use line wrapping together with the command history feature to recall and modify previous complex command entries.

Deleting Entries

Use any of these keystrokes to delete command entries if you make a mistake or change your mind:

Keystrokes	Task
Press **Delete** or **Backspace**.	Erase the character to the left of the cursor.
Press **Ctrl-D**.	Delete the character at the cursor.
Press **Ctrl-K**.	Delete all characters from the cursor to the end of the command line.
Press **Ctrl-U** or **Ctrl-X**.	Delete all characters from the cursor to the beginning of the command line.
Press **Ctrl-W**.	Delete the word to the left of the cursor.
Press **Esc D**.	Delete from the cursor to the end of the word.

Scrolling Down a Line or a Screen

When you use the help facility to list the commands available in a particular mode, the list is often longer than the terminal screen can display. In such cases, a More prompt appears at the bottom of the screen. To respond to the More prompt, use these keystrokes:

Keystrokes	Task
Press **Return**.	Scroll down one line.
Press **Space**.	Scroll down one screen.
Press **Esc**.	Stop scrolling and return to the main prompt.

Redisplaying the Current Command Line

If you enter a command and a message appears on your screen, you can easily recall your current command line entry. To do so, use these keystrokes:

Keystrokes	Task
Press **Ctrl-L** or **Ctrl-R**.	Redisplay the current command line.

Transposing Mistyped Characters

If you have mistyped a command entry, you can transpose the mistyped characters by using these keystrokes:

Keystrokes	Task
Press **Ctrl-T**.	Transpose the character to the left of the cursor and the character located at the cursor.

Controlling Capitalization

You can capitalize or lowercase words or capitalize a set of letters with these keystrokes:

Keystrokes	Task
Press **Esc C**.	Capitalize at the cursor.
Press **Esc L**.	Change the word at the cursor to lowercase.
Press **Esc U**.	Capitalize letters from the cursor to the end of the word.

Designating a Keystroke as a Command Entry

To use a particular keystroke as an executable command, insert a system code for this purpose:

Keystrokes	Task
Press **Ctrl-V** or **Esc Q**.	Insert a code to indicate to the system that the keystroke that follows should be treated as a command entry, *not* an editing key.

Disabling Enhanced Editing Mode

To disable enhanced editing mode and revert to the editing mode, use this command in privileged EXEC mode:

Command	Task
`DSLAM#terminal no editing`	Disable the enhanced editing features for the local line.

If you have prebuilt scripts that do not interact well when enhanced editing is enabled, you can disable enhanced editing mode. To reenable enhanced editing mode, use the **terminal editing** command

Ending a Session

After you use the **setup** command or other configuration command, exit the DSLAM and quit the session.

To end a session, use this EXEC command:

Command	Task
DSLAM>**quit**	End the session.

CHAPTER 2

Configuring Terminal Lines and Modem Support

This chapter describes how to configure lines, modems, and terminal settings to access the ATM switch for management purposes. The Cisco DSLAM has two types of terminal lines:

- A console line
- An auxiliary line

Most line setup is the same for all types of lines, but certain commands, such as those having to do with modem control, apply only to the auxiliary port.

This chapter includes these sections:

Configuring Terminal Lines

Configuring terminal lines is a two-step process:

Step 1 Set up the lines for the terminals or other asynchronous devices attached to them.

Step 2 Configure the parameters for each line.

The tasks involved in these steps are described in these sections:

Preparing to Configure Lines

Use line configuration mode to enter line configuration commands that affect a specified console, auxiliary, or virtual terminal line. To enter line configuration mode, use this command in global configuration mode:

Command	Task
DSLAM(config)#**line** [**aux** | **console** | **vty**] *line-number* [*ending-line-number*]	Specify an auxiliary, console, or virtual terminal line to configure.

The terminal from which you locally configure the system is attached to the console port.

Example

This example specifies the console port and begins line configuration mode:

```
DSLAM(config)# line con 0
DSLAM(config-line)#
```

The auxiliary port supports modem connections. See the "Setting Up Modem Control on the Auxiliary Port" section on page 2-5, to set up modem support on the auxiliary port.

Configuring the console port or virtual terminal lines allows you to specify communication parameters and autobaud connections, and configure terminal operating parameters for the terminal you are using. These tasks are described in the "Defining Terminal Operation Characteristics" section on page 2-13.

You can also use the line command to create virtual terminal lines. This example shows how to create and configure the maximum 4 virtual terminal lines with the "no login" feature:

```
DSLAM(config)#line vty 0 4
DSLAM(config-line)#no login
```

Setting Communication Parameters

You can change the default parameters for terminal communications to meet the requirements of the terminal or host to which you are attached. To do so, use one or more of these commands in line configuration mode:

Command	Task
speed *bps* **txspeed** *bps* **rxspeed** *bps*	Set the line speed. Choose from line speed, transmit speed, or receive speed. Speed applies to the auxiliary port only.
databits {**5** | **6** | **7** | **8**}	Set the data bits.
stopbits {**1** | **1.5** | **2**}	Set the stop bits.
parity {**none** | **even** | **odd** | **space** | **mark**}	Set the parity bit.

This example shows how to configure the auxiliary line with a speed of 19,200 bits per second (bps):

```
DSLAM(config)#line aux 0
```

```
DSLAM(config-line)#speed 19200
```

Configuring Automatic Baud Detection

You can configure a terminal to automatically detect the baud rate over an asynchronous serial line. This configuration applies to the auxiliary port only.

To set up automatic baud detection, use this command in line configuration mode:

Command	Task
`DSLAM(config-line)#autobaud`	Set the terminal to automatically detect the baud rate.

To start communications using automatic baud detection, press **Return** multiple times at the terminal:

- Press **Return** three times for a 600-, 1800-, or 19200-baud line to detect the baud rate
- Press **Return** two times to set up a line at any other baud rate
- Press **Return** after the baud rate is detected, and the EXEC displays another system prompt

Changing the Default Privilege Level for Lines

To change the default privilege level for a given line or group of lines, use this command in line configuration mode:

Command	Task
`DSLAM(config-line)#privilege level` *level*	Specify a default privilege level for a line.

Allowable values for the privilege level are *0* through *15*.

Configuring Flow Control for Communication

On the auxiliary port, you can set both hardware and software flow control between the DSLAM and the devices attached to it.

To configure flow control between the DSLAM and attached device, use one or more of this commands in line configuration mode:

Command	Task
flowcontrol {**none** | **software** [**in** | **out**] | **hardware** [**in** | **out**]}	Set the terminal flow control.
start-character *ascii-number*	Set the flow control start character.
stop-character *ascii-number*	Set the flow control stop character.

Allowable values for the **start-character** and **stop-character** commands are *CHAR* or *0* through *255*.

Both software and hardware flow control are bidirectional. If you do not specify a direction, the DSLAM enables software flow control in both directions. For information about setting up hardware flow control on the EIA/TIA-232 line, see the hardware installation and maintenance manual for your product.

Defining a Command String for Automatic Execution

You can define a command that automatically executes upon connection to another host. Any appropriate EXEC command and any switch or host name that occurs with the EXEC command is allowed. To do so, use this command in line configuration mode:

Command	Task
`DSLAM(config-line)#autocommand` *command*	Define a command string to be automatically executed.

Specifying the Transport Protocol for a Specific Line

You can specify the protocols for individual lines by setting the protocol for incoming and outgoing connections and changing the default (preferred) protocol for a line. The default transport protocol is Telnet.

To specify transport protocols, use one or more of these commands in line configuration mode:

Command	Task
transport input {**all** | **telnet** | **none**}	Define which protocols can connect to a specific line of the DSLAM.
transport output {**all** | **telnet** | **none**}	Determine the protocols for outgoing connections from a line.
transport preferred {**all** | **telnet** | **none**}	Specify the protocol to use if the user did not specify one.
transport preferred none	Prevent errant connection attempts.

The system accepts a host name entry at the EXEC system prompt as a Telnet command. If you incorrectly type the host name, the system interprets the entry as an incorrect Telnet command and displays an error message indicating that the host does not exist. The **transport preferred none** command disables this option if you incorrectly type a command at the EXEC prompt, and the system does not attempt to make a Telnet connection.

Establishing Terminal Session Limits

You can set a time limit on a terminal session. To limit terminal sessions, use this commands in line configuration mode:

Command	Task
`session-timeout` *minutes* `[output]`	Set the idle session timeout interval.

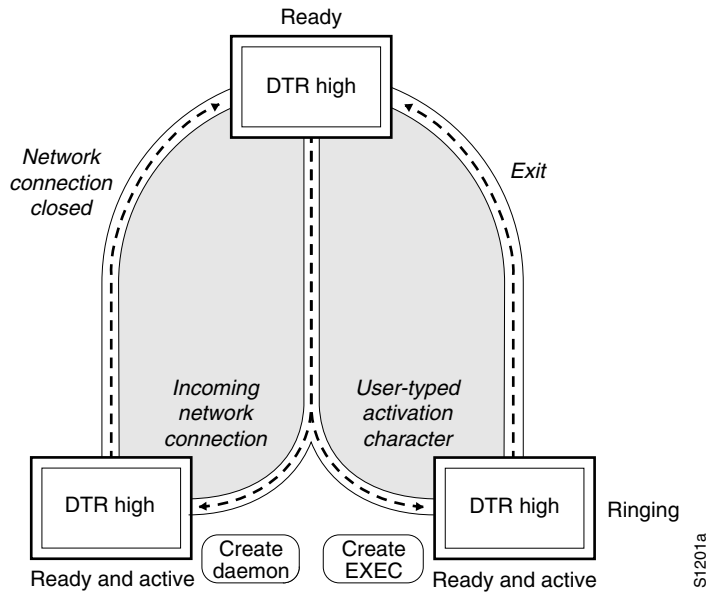
Setting Up Modem Control on the Auxiliary Port

This section describes modem control and how to configure it on the modem port. These subsections are included:

Modem Control Process

Figure 2-1 illustrates how modem control works on the DSLAM auxiliary port.

Figure 2-1 EXEC and Daemon Creation on a Line with No Modem Control



These figures show two processes:

- The *create daemon* process, used to create a TTY daemon that handles the incoming network connection
- The *create EXEC* process, used to create the process that interprets user commands.

In the figures, the current signal state and the signal line are listed inside each box. The state of the line is listed next to the box. (You can display the current state of a line with the **show line** command.) Events that change that state appear in italics along the event path, with the software actions described within the ovals.

Figure 2-1 illustrates line behavior when no modem control is set. The data terminal ready (DTR) output is always high, and CTS and RING are ignored. The DSLAM creates an EXEC when the activation character is typed. Incoming Transmission Control Protocol (TCP) connections occur instantly if the line is not in use and can be closed only by the remote host.

Configuring Automatic Dialing

With the dialup capability, you can set a modem to automatically dial the phone number of a remote DSLAM. This feature offers cost savings because phone line connections are made as needed. You pay for using the phone line only when there is data to be received or sent. To configure a line for automatic dialing, use this command in line configuration mode:

Command	Task
`DSLAM(config-line)#modem dtr-active`	Configure a line to initiate automatic dialing.

Automatically Answering a Modem

You can configure a line to automatically answer a modem. You also configure the modem to do this:

- Answer the telephone automatically if DTR is high.
- Drop connections when DTR is low.
- Use its Carrier Detect (CD) signal to accurately reflect the presence of carrier.

Note Configuring the modem is a modem-dependent process.

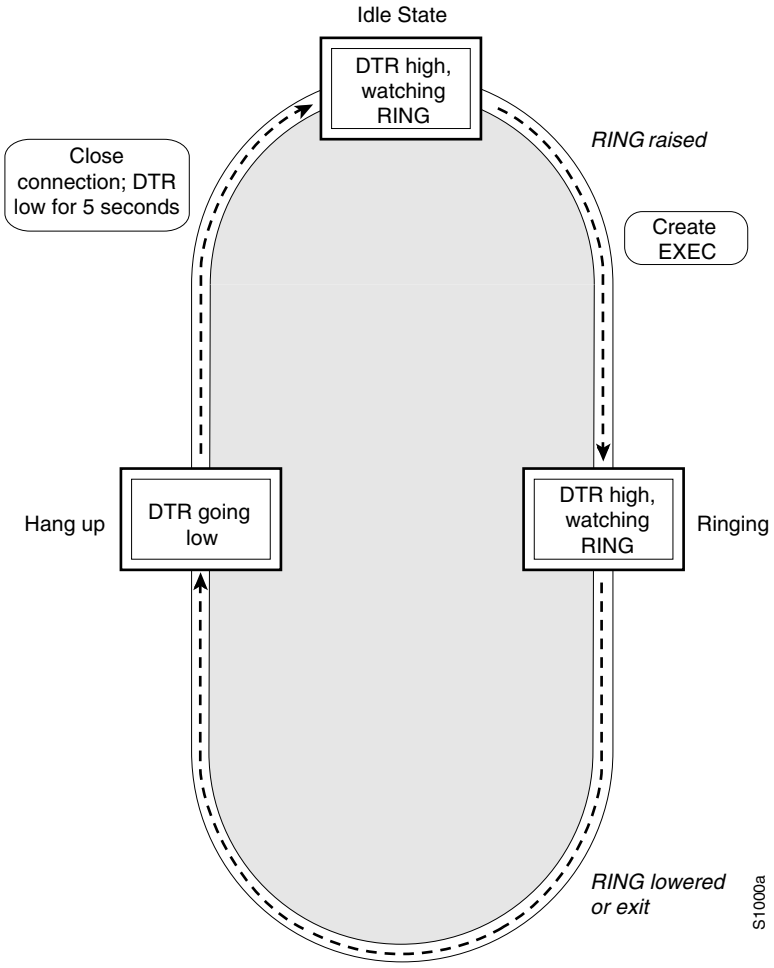
Wire the modem's CD signal (generally pin 8) to the RING input (pin 22) of the DSLAM, then use this command in line configuration mode:

Command	Task
`DSLAM(config-line)#modem dialin`	Configure a line to automatically answer a modem.

You can turn on the modem's hardware flow control independently to act on the status of the DSLAM's clear to send (CTS) input. Wire CTS to the signal the modem uses for hardware flow control. If the modem is set to control hardware flow in both directions, you may also need to wire the modem's flow control input to a signal that the DSLAM always sets to high (such as DTR).

Figure 2-2 illustrates the modem dialin process. When the DSLAM detects a signal on the RING input of an idle line, it starts an EXEC or autobaud process on that line. If the RING signal disappears on an active line, the DSLAM closes any open network connections and terminates the EXEC. If the user exits the EXEC or the DSLAM terminates it because of no user input, the line hangs up the modem by lowering the DTR signal for 5 seconds. After 5 seconds, the modem is ready to accept another call.

Figure 2-2 EXEC Creation on a Line Configured for a High-Speed Dial-Up Modem



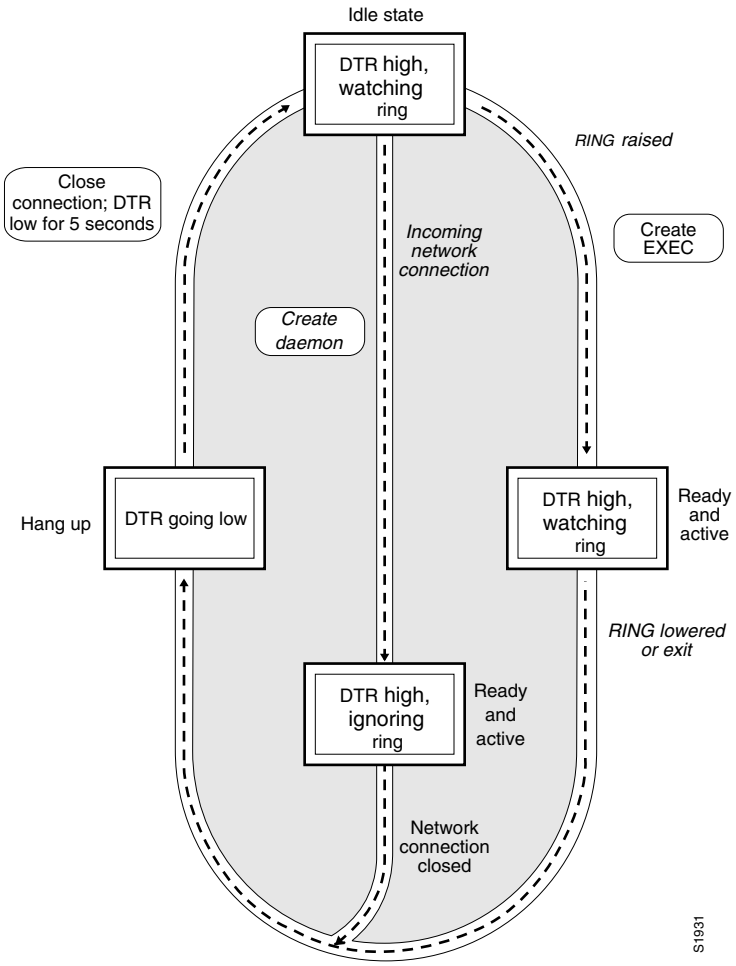
Supporting Dial-In and Dial-Out Modems

To configure a line for both incoming and outgoing calls, use this command in line configuration mode:

Command	Task
`DSLAM (config-line)#modem inout`	Configure a line for both incoming and outgoing calls.

Figure 2-3 illustrates the modem in-out process.

Figure 2-3 EXEC and Daemon Creation on a Line Configured for Incoming and Outgoing Calls



If the line is activated by:

- Raising RING, it behaves exactly as a line configured with the **modem dialin** subcommand.
- An incoming TCP connection, the line behaves similarly to a non-modem line.

Note If your system uses dial-out modems, consider using access lists to prevent unauthorized use.

Configuring a Line Timeout Interval

You can change the interval that the DSLAM waits for CTS after raising DTR in response to RING from the default of 15 seconds. To do so, use this command in line configuration mode:

Command	Task
`modem answer-timeout` *seconds*	Configure modem line timing.

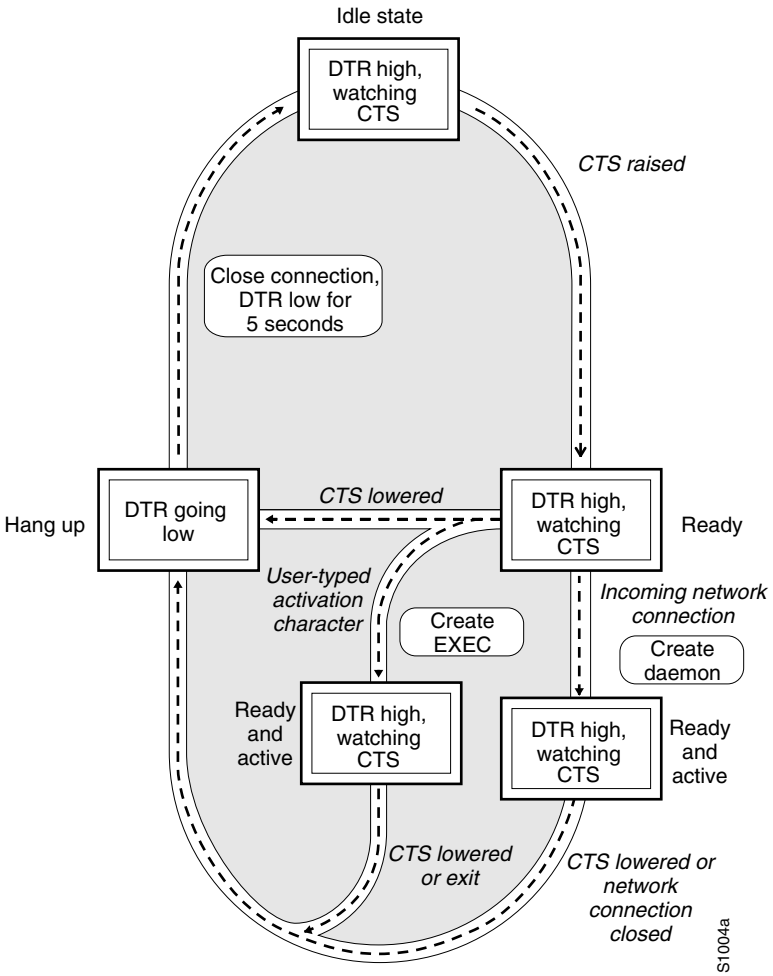
Closing Modem Connections

You can configure a line to close all connections from a user's terminal when the terminal is turned off, and prevent inbound connections to devices that are out of service. To do so, use this command in line configuration mode:

Command	Task
`modem printer`	Configure a line to close all connections.

Figure 2-4 illustrates the modem printer process requirement for a high CTS throughout the use of the line.

Figure 2-4 EXEC and Daemon Creation on a Line Configured for Continuous CTS



If CTS is not high, the user's typed input is ignored and incoming connections are refused (or stepped to the next line in a rotary group).

A DSLAM can reliably detect a CTS signal change if the signal remains in the new state for at least one full second.

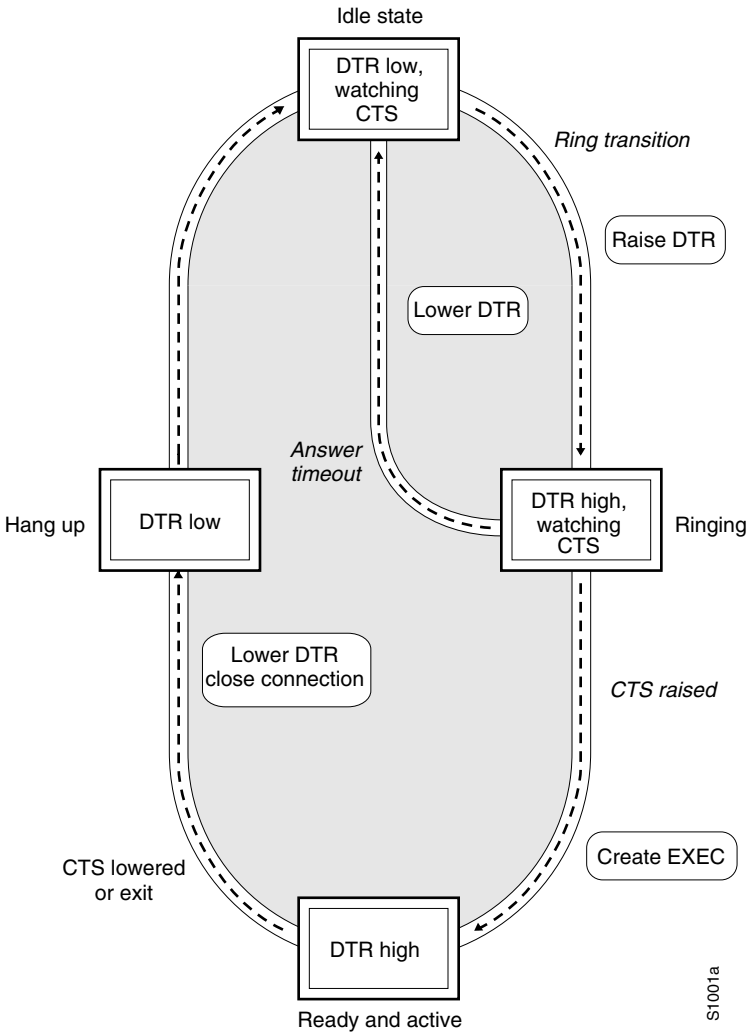
Supporting Old-Style Dial-In Modems

The DSLAM supports dial-in modems that use DTR to control the off-hook status of the telephone line. To do this, use this command while in line configuration mode:

Command	Task
`modem callin`	Configure a line for a dial-in modem.

Figure 2-5 illustrates the modem call in process.

Figure 2-5 EXEC Creation on a Line Configured for Modem Callin



When a modem dialing line is idle, the DTR is in a low state and waits for a transition to occur on the RING input. This transition causes the line to raise DTR and start watching the CTS signal from the modem. After the modem raises CTS, the DSLAM creates an EXEC on the line. If the timeout interval (set with the **modem answer-timeout** command) expires before the modem raises CTS, the line lowers DTR and returns to the idle state

Note The modem callin and modem printer line configuration commands ensure that when the line is hung up or CTS drops, the line reverts from SLIP mode to normal interactive mode. These commands do not work if you use the async dedicated command to put the line in network mode permanently.

Although you can use the modem callin line configuration command with newer modems, the modem dialin line configuration command described earlier in this section is more appropriate. The modem dialin command frees up CTS for hardware flow control. Modern modems do not require the DTR to take a phone line off-hook.

Configuring Rotary Groups

You can make connections to the next free line in a group of lines, also called a rotary or hunt group. A line can be in only one rotary group. A rotary group can consist of a single line or several contiguous lines. The console line (line 0) cannot be in a rotary group.

If you want to assign the rotary as the single auxiliary port line because the auxiliary port is not necessarily the same line number on all hardware. By assigning the line to a rotary group, you do not have to track the actual line number. Another reason to use a rotary group is that if the device supports local area transport (LAT), an inbound service can only be bound to a rotary group. It cannot be bound to a port number.

To configure a rotary group, use this command in line configuration mode:

Command	Task
rotary *group*	Add a line to the specified rotary group.

Configuring High-Speed Modem Support

Modems that operate over normal dial-up telephone lines at speeds of 9600 bits per second (bps) and higher do not guarantee a specific throughput; instead, they operate at a speed that depends on the quality of the line, the effectiveness of data compression algorithms on the data being transmitted, and other variables. These modems use hardware flow control to stop the data from reaching the host by toggling an EIA/TIA-232 signal when they cannot accept any more data.

In addition to hardware flow control, dial-up modems require special software handling. You must configure the modems to:

- Create an EXEC when a user dials in.
- Hang up when the user exits the EXEC.
- Close any existing network connections if the telephone line hangs up in the middle of a session.

The DSLAM supports hardware flow control on its CTS input, which is also used by the normal modem handshake. To configure and use a high-speed modem, perform these tasks, beginning in line configuration mode:

Step	Command	Task
1	DSLAM(config-line)#**flowcontrol hardware**	In line configuration mode, enable outgoing hardware flow control based on the CTS input.
2	DSLAM(config-line)#**end**	Enter privileged EXEC command mode.
3	DSLAM#**debug modem**	Display informational messages on the console terminal about modem control events, such as signal transitions and autobaud progress.
4	DSLAM#**show line**	Display the status of a line. In the detailed command output, a Status line with "Idle" identifies inactive modem dialin lines and all other modem lines; a Status line with "Ready" identifies lines in use.

Supporting Reverse TCP Connections

The DSLAM can receive incoming connections on the auxiliary port. This capability allows you to attach serial printers, modems, and other shared peripherals to the DSLAM and drive them remotely from other systems. The DSLAM supports reverse TCP connections.

Front-Ending

The specific TCP port or socket to which you attach the peripheral device determines the type of service the DSLAM provides on that line. When you attach the serial lines of a computer system or a data terminal switch to the auxiliary port of the DSLAM, the DSLAM acts as a network front end for a host that does not support the TCP/IP protocols. This arrangement is sometimes called *front-ending* or *reverse connection mode*.

To connect the auxiliary port, the remote host or terminal must specify a particular TCP port on the DSLAM. If Telnet protocols are required, that port is 2000 (decimal) plus the decimal value of the line number.

TCP Streams

If a raw TCP stream is required, the port is 4000 (decimal) plus the decimal line number. The raw TCP stream is usually the required mode for sending data to a printer.

The Telnet protocol requires that carriage return characters be translated into carriage return and line feed character pairs. You can turn this translation off by specifying the Telnet binary mode option. To specify this option, connect to port 6000 (decimal) plus the decimal line number.

Defining Terminal Operation Characteristics

In line configuration mode, you can set terminal operation characteristics for that line until you change the line parameters.

You can temporarily change the line settings using the **terminal** EXEC commands described in the Chapter 1, "Cisco DSLAM User Interface."

Define the terminal operation characteristics by performing the tasks in these sections:

Specifying the Terminal Type

You can specify the type of terminal connected to a line. This feature has two benefits: it records the type of terminal attached to a line, and it can inform the remote host of the terminal type for display management. To specify the terminal type, use this command in line configuration mode:

Command	Task
terminal-type *terminal-name*	Specify the terminal type.

Setting the Terminal Screen Length and Width

By default, the DSLAM provides a screen display of 24 lines by 80 characters. You can reconfigure these values if they do not meet the needs of your terminal by performing these tasks in line configuration mode:

Step	Command	Task
1	**length** *screen-length*	Set the screen length.
2	**width** *characters*	Set the screen width.

The values set can be learned by some host systems that use this type of information in terminal negotiation. Set a value of 0 for the screen length to disable pausing between windows of output.

Defining Escape Character and Other Key Sequences

You can define or modify the default key sequences to execute functions for system escape, terminal activation, disconnect, and terminal pause. To define or change the default sequence, use one or more of these commands in line configuration mode:

Command	Task
escape-character *ascii-number*	Change the system escape sequence. The escape sequence indicates that the codes that follow have special meaning. The default sequence is Ctrl-^.
activation-character *ascii-number*	Define a session activation sequence or character. Typing this sequence at a vacant terminal begins a terminal session. The default key is Return.
disconnect-character *ascii-number*	Define the session disconnect sequence or character. Typing this sequence at a terminal ends the session with the DSLAM. There is no default sequence.
hold-character *ascii-number*	Define the hold sequence or character that causes output to the terminal screen to pause. There is no default sequence. To continue the output, type any character after the hold character. To use the hold character in normal communications, precede it with the escape character.

Note If you are using the **autoselect** command, do not change the activation character from the default value of Return. If you change this default, **autoselect** may not function immediately.

Specifying the International Character Display

You can use a 7-bit character set (such as ASCII) or you can enable a full 8-bit international character set (such as ISO 8859) to allow special graphical and international characters for use in banners and prompts, and to add special characters such as software flow control. You can configure these settings globally by interface or locally at the user level. Use these criteria for determining the configuration mode to use when setting up this feature:

- If a large number of connected terminals support non-default ASCII bit settings, use the global configuration commands.
- If only a few of the connected terminals support non-default ASCII bit settings, use line configuration commands or the EXEC local terminal setting commands.

Note Setting the EXEC character width to 8 bits can cause failures. If a user on a terminal that is sending parity enters the **help** command, an unrecognized command message appears because the system is reading all 8 bits, although the eighth bit is not needed for the **help** command.

To specify a character set on a global basis, use *one or both* of these commands in global configuration mode:

Command	Task
default-value exec-character-bits {**7** | **8**}	Specify the character set used in EXEC and configuration command characters.
default-value special-character-bits {**7** | **8**}	Specify the character set used in special characters such as software flow control, hold, escape, and disconnect characters.

To specify a character set based on hardware or software, or on a per-line basis, use the appropriate command in line configuration mode:

Command	Task
databits {**5** | **6** | **7** | **8**}	Set the number of databits per character that are generated and interpreted by hardware.
data-character-bits {**7** | **8**}	Set the number of databits per character that are generated and interpreted by software.
exec-character-bits {**7** | **8**}	Specify the character set used in EXEC and configuration command characters on a per-line basis.
special-character-bits {**7** | **8**}	Specify the character set used in special characters such as software flow control, hold, escape, and disconnect characters on a per-line basis.

Note If you are using the **autoselect** function, let the activation character default to Return, and **exec-character-bits** default to 7. If you change these defaults, the application does not recognize the activation request.

Setting Character Padding

You can change the character padding on a specific output character. Character padding adds a number of null bytes to the end of the string and can make a string conform to an expected length. To set character padding, use this command in line configuration mode:

Command	Task
padding *ascii-number count*	Set padding, *count*, on a specific output character, *ascii-number,* for the specified line.

Disabling Enhanced Editing Mode

To disable enhanced editing mode and revert to the editing mode of earlier software releases, use this command in line configuration mode:

Command	Task
no editing	Disable the enhanced editing features for a particular line.

You can disable enhanced editing if you have prebuilt scripts that do not interact well when enhanced editing is enabled. You can reenable enhanced editing mode using the **editing** command.

Providing Line Connection Information after the Login Prompt

You can provide the host name, line number, and location each time an EXEC is started or an incoming connection is made. The line number banner appears immediately after the EXEC banner or incoming banner. It is useful for tracking problems with modems because it lists the host and line for the modem connection. Modem information is also included if applicable.

To provide service line number information, use this command in global configuration mode:

Command	Task
service *linenumber*	Provide service line number information after the EXEC or incoming banner.

Enabling Password Checking at Login

You can enable password checking on a particular line so that the user is prompted to enter a password at the system login screen. You must then also specify a password. To do so, perform these tasks in line configuration mode:

Step	Command	Task
1	**login**	Enable password checking on a per-line basis using the password specified with the **password** command.
2	**password** *password*	Assign a password to a particular line.

You can enable password checking on a per-user basis so that authentication is based on the user name specified with the **username** global configuration command. To enable this type of password checking, use one of these commands in line configuration mode:

Command	Task
login local	Enable password checking on a per-user basis using the user name and password specified with the **username** global configuration command.
login tacacs or **login authentication** {**default** | **list-name**}	Select the TACACS-style user ID and password-checking mechanism.

Use the **login tacacs** command with Terminal Access Controller Access Control System (TACACS) and extended TACACS Plus. Use the **login authentication** command with authentication, authorization, and accounting (AAA)/TACACS+.

By default, virtual terminals require passwords. If you do not set a password for a virtual terminal, it responds to attempted connections by displaying an error message and closing the connection. Use the **no login** command to disable this behavior and allow connections without a password.

Checking Password Examples

This example shows password checking enabled for a virtual terminal line 1:

```
DSLAM(config)# line vty 1
DSLAM(config-line)# login
DSLAM(config-line)# password letmein
```

This example shows password checking enabled on a per-user basis:

```
DSLAM(config)# username jksmith password 0 letmein
DSLAM(config)# username lmjones password 0 littlerock
DSLAM(config)# line vty 1
DSLAM(config-line)# login local
```

Configuring Terminal Banner Messages

These sections explain how to configure terminal messages:

Configuring a Message-of-the-Day Banner

You can configure a message-of-the-day (MOTD) banner to display on all connected terminals. This message is displayed at login and is useful for sending messages that affect all network users, such as impending system shutdowns. To do so, use this command in global configuration mode:

Command	Task
banner motd *c message c*	Configure a message-of-the-day banner.

Configuring a Line Activation Message

You can configure a line activation message to display when an EXEC process such as line activation or an incoming connection to a virtual terminal is created. To do so, use this command in global configuration mode:

Command	Task
banner exec *c message c*	Configure a message to be displayed on terminals with an interactive EXEC.

Configuring an Incoming Message Banner

You can configure a message to display on terminals connected to reverse Telnet lines. This message is useful for providing instructions to users of these types of connections. Reverse Telnet connections are described in more detail in the "Supporting Reverse TCP Connections" section on page 2-13.

To configure the message that will be sent on incoming connections, use this command in global configuration mode:

Command	Task
banner incoming *c message c*	Configure messages to display on terminals connected to reverse Telnet lines.

Configuring an Idle Terminal Message

You can configure messages to display on a console or terminal that is not in use. Also called a *vacant message*, this message is different from the banner message displayed when an EXEC process is activated. To configure an idle terminal message, use this command in line configuration mode:

Command	Task
vacant-message *c message c*	Display an idle terminal message.

Enabling or Disabling the Display of Messages

You can control display of the MOTD and line activation banners. By default, these banners display on all lines. To suppress or resume these messages, use one of these commands in line configuration mode:

Command	Task
no exec-banner	Suppress banner display.
exec-banner	Resume the display of the EXEC or MOTD banners.

Banner Message Example

This example shows how to use the **banner** global configuration command and **no exec-banner** line configuration command to notify your users that the server will be reloaded with new software:

```
DSLAM(config)# banner exec /
```

```
Enter TEXT message.  End with the character '/'.

Unauthorized access prohibited./
DSLAM(config)# banner incoming /
You are connected to a Hayes-compatible modem.

Enter the appropriate AT commands.
Remember to reset anything to change before disconnecting.
/
DSLAM(config)# banner motd /
The switch will go down at 6pm for a software upgrade.
/
DSLAM(config)# line vty 0 4
DSLAM(config-line)# no exec-banner
DSLAM(config-line)#
```

CHAPTER 3

Initially Configuring the Cisco DSLAM

This chapter describes how to initially configure the Cisco DSLAMs, and includes these sections:

Methods for Configuring the DSLAM

The DSLAM default configuration is suitable for operation with most networks. By using network management applications and the text-based command-line interface (CLI), you can configure and customize all aspects of DSLAM operation to suit your needs.

The DSLAM ships with the ATM address autoconfigured, allowing the DSLAM to:

- Automatically configure attached end systems using the Interim Local Management Interface (ILMI) protocol
- Establish itself as a node in a single-level Private Network-Network Interface (PNNI) routing domain.

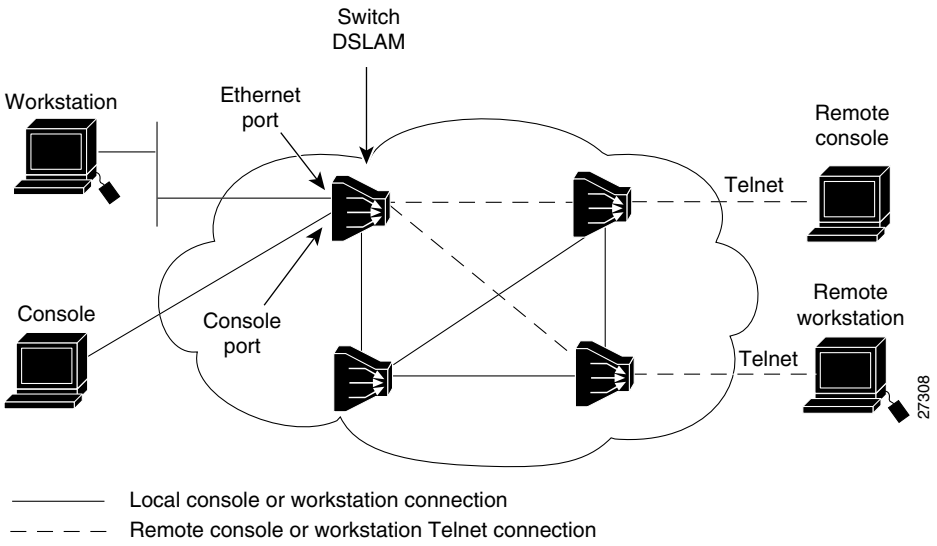
The ILMI and PNNI protocols allow the DSLAM to be entirely self-configured when you use these protocols with an IP address autoconfiguration mechanism such as BOOTP.

You must assign an IP address to allow up to eight simultaneous Telnet sessions to connect to the DSLAM or to use the Simple Network Management Protocol (SNMP) system for the DSLAM. The Ethernet IP address is assigned either manually or by a BOOTP server. See the "Configuring IP Interface Parameters" section on page 3-12.

You can use either of two methods for configuring a DSLAM (Figure 3-1):

- From a local console or workstation—Connect to the console port or connect to the Ethernet port of a DSLAM. This connection allows you to issue CLI commands directly to the DSLAM chassis.
- From a remote console or workstation—Initiate a Telnet connection to a target DSLAM. Telnet allows you to remotely issue CLI commands to that chassis.

Figure 3-1 Two Methods of Configuring a DSLAM



Port and Slot Configuration

The DSLAM contains an NI-2 card and up to 34 line (modem) cards depending on the DSLAM. The slot configurations on the different DSLAMs are as follows:

- Cisco 6015
  - six line card slots
  - one NI-2 card slot
- Cisco 6100
  - 32 line card slots
  - two NI-2 card slots (only one slot active)

- Cisco 6130
  - 32 line card slots
  - two NI-2 card slots (to provide redundancy)
- Cisco 6160
  - 32 line card slots
  - two NI-2 card slots (to provide redundancy)
- Cisco 6260
  - 30 line card slots
  - two NI-2 card slots (to provide redundancy)

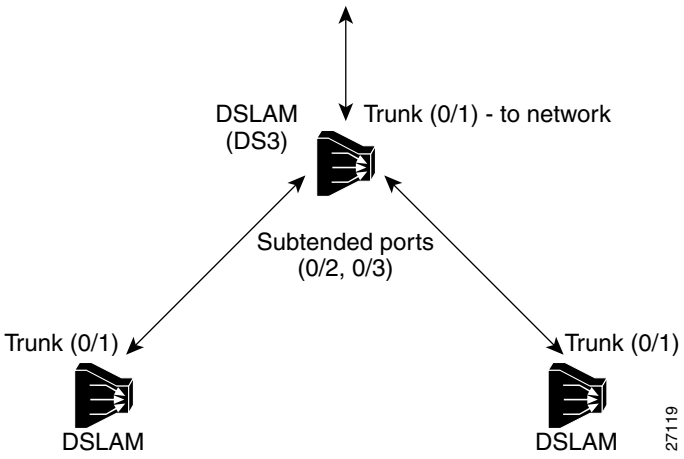
In all the chassis, the NI-2 card handles the network interfaces. The NI-2 card has either OC3 or DS3 interfaces.

Line cards are assigned ports 1 to 4 or 1 to 8 in consecutive slots. Table 3-1 lists NI-2 port assignments. Figure 3-2 shows the port connection arrangement.

Table 3-1 NI-2 Port Assignments

Port Type	OC3 Configuration Assigned slot/port	DS3 Configuration Assigned slot/port	Function
Switch, Ethernet	0/0	0/0	The ATM switch or Ethernet CPI port (internal).
Trunk	0/1	0/1	The trunk port connects to the network, either directly or through a subtended port in another DSLAM.
Subtend 1	0/2	0/2	A subtended port connects a second DSLAM to the network through a primary DSLAM. See Figure 3-3.
Subtend 2	N/A	0/3	The DS3 configuration has a second subtended port.

Figure 3-2 DSLAM Port Connections



Configuration Prerequisites

Obtain this information before you configure your DSLAM:

- To configure a BOOTP server to inform the DSLAM of its Ethernet IP address and mask, you need the Media Access Control (MAC) address of the Ethernet port.
- To configure a new ATM address for the DSLAM (an autoconfigured ATM address is assigned by Cisco), you need an ATM address assigned by your system administrator.
- If you are not using BOOTP, obtain an IP address and a subnet mask.

Verifying Installed DSLAM Software and Hardware

When you first power on your console and DSLAM, a screen similar to this appears:

```
              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706
```

The script then displays the banner information, including the software version, followed by the installed hardware configuration.

```
cisco ASP1 (R4600) processor with 16384K bytes of memory.

Cisco Internetwork Operating System Software
IOS (tm) PNNI Software (LS-WP-M), Version XX.X(X.X.WAX.X.XX)
Copyright (c) 1986-1998 by cisco Systems, Inc.
Compiled Tue 11-Jan-98 02:59 by
Image text-base: 0x600108D0, data-base: 0x603EE000

8192K bytes of Flash internal SIMM (Sector size 256K).


Press RETURN to get started!
```

The DSLAM should now be operating correctly and transferring data.

Configuring the BOOTP Server

The BOOTP protocol automatically assigns an Ethernet IP address by adding the MAC and IP addresses of the Ethernet port to the BOOTP server configuration file. When the DSLAM boots, it automatically retrieves the IP address from the BOOTP server.

The DSLAM performs a BOOTP request *only* if the current IP address is set to 0.0.0.0. (This is the default for a new DSLAM or a DSLAM that has had its configuration file cleared using the **erase startup-config** command.)

To allow the DSLAM to retrieve its IP address from a BOOTP server you must first determine the MAC address of the DSLAM and then add that MAC address to the BOOTP configuration file on the BOOTP server.

Complete the following tasks to create a BOOTP server configuration file:

Step 1 Install the BOOTP server code on the workstation, if it is not already installed.

Step 2 Determine the MAC address from the label on the chassis.

Step 3 Add an entry in the BOOTP configuration file (usually */usr/etc/bootptab*) for each DSLAM. Press **Return** after each entry to create a blank line between each entry. See the sample BOOTP configuration file that follows this table.

Step 4 Restart the DSLAM to automatically request the IP address from the BOOTP server.

Example

This example BOOTP configuration file shows the newly added DSLAM entry:

```
# /etc/bootptab: database for bootp server (/etc/bootpd)
#
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
#       first field -- hostname
#                       (may be full domain name)
#
#       hd -- home directory
#       bf -- bootfile
#       cs -- cookie servers
#       ds -- domain name servers
#       gw -- gateways
#       ha -- hardware address
#       ht -- hardware type
#       im -- impress servers
#       ip -- host IP address
#       lg -- log servers
#       lp -- LPR servers
#       ns -- IEN-116 name servers
#       rl -- resource location protocol servers
#       sm -- subnet mask
#       tc -- template host (points to similar host entry)
#       to -- time offset (seconds)
#       ts -- time servers

<display truncated>

###############################################################################
# Start of individual host entries
###############################################################################
Switch:         tc=netcisco0:   ha=0000.0ca7.ce00:      ip=192.31.7.97:
dross:          tc=netcisco0:   ha=00000c000139:        ip=192.31.7.26:

<information deleted>
```

Setting the Subtend Node Identifier

In a subtended network configuration, the subtend node acts as the host node connecting all the nodes to the network. This node is identified to the network using the **subtend-id** command.

To set the subtend node identifier, use the following command:

Command	Task
DSLAM# **subtend-id** *node#*	In priviledged EXEC mode, identify *node#* as the subtend host node.

Example

This example sets the DSL subtend node identifier to node 12:

```
DSLAM> enable
Password:
DSLAM# subtend-id 12
```

Configuring the ATM Address

The DSLAM is autoconfigured with an ATM address using a hierarchical addressing model similar to the OSI network service access point (NSAP) addresses. PNNI uses this hierarchy to construct ATM peer groups. ILMI uses the first 13 bytes of this address as the switch prefix that it registers with end systems.

Note If you manually change an ATM address, you must maintain the uniqueness of the address across the network.

Configuring ATM Addressing

This section describes the ATM addressing scheme and tells you how to

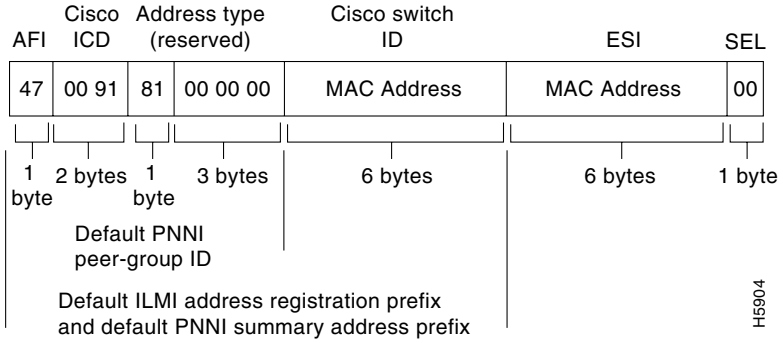
- Use the ATM default addressing scheme
- Manually set ATM addresses

Using the ATM Default Addressing Scheme

This section describes the default addressing scheme and the features and implications of using this scheme.

During the initial startup, the DSLAM generates an ATM address using the defaults shown in Figure 3-3.

Figure 3-3 ATM Address Format Defaults



The default addressing scheme includes:

- Authority and format identifier (AFI)—1 byte
- Cisco specific International Code Designator (ICD)—2 bytes
- Cisco specific information—4 bytes
- Cisco switch ID—6 bytes (used to distinguish multiple switches). The first 13 bytes of the address is a switch prefix used by ILMI in assigning addresses to end stations connected to User-Network Interface (UNI) ports.
- MAC address of the switch—6 bytes (used to distinguish multiple end system identifier [ESI] addresses). Both the DSLAM ID and ESI MAC address fields in the ATM address are the same, but they may not be the same as the address printed on the chassis label. Use the ATM address fields when you configure the ATM addressing scheme.
- Selector (SEL) equals 0—1 byte

If you use the default address format, these features and implications apply:

- The default address format enables you to manually configure other switches to be used in a single-level PNNI routing domain consisting primarily of autoconfigured Cisco ATM switches. You must use a globally unique MAC address to generate the ATM address.
- You can assign the same MAC address for bytes 8 through 13 and bytes 14 through 19.
- To achieve scalable ATM routing, you need two addresses when you connect to a large ATM network with multiple levels of PNNI hierarchy.
- Do not use summary addresses with fewer than 13 bytes with autoconfigured ATM addresses. Other switches with autoconfigured ATM addresses that match the DSLAM summary can exist outside of the default peer group.

Manually Setting the ATM Address

You can configure a new ATM address that replaces the previous ATM address when running IISP software only, or that replaces the previous ATM address and generates a new PNNI node ID and peer group ID as follows:

- To configure a new ATM address that replaces the previous ATM address when running IISP software only, see the *ATM Switch Router Software Configuration Guide, Chapter 10.*

  http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/ilmi_cnf.htm
- To configure a new ATM address that replaces the previous ATM address and generates a new PNNI node ID and peer group ID, see the *ATM Switch Router Software Configuration Guide, Chapter 11.*

http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_1/lhouse/sw_confg/access.htm

You can configure multiple addresses for a single switch and use this configuration during ATM address migration. ILMI registers end systems with multiple prefixes during this period until you remove an old address. PNNI automatically summarizes all the switch prefixes in its reachable address advertisement.

For operation with ATM addresses other than the autoconfigured ATM address, use the **atm address** command to manually assign a 20-byte ATM address to the switch. The **atm address** command *address_template* variable can be a full 20-byte address or a 13-byte prefix followed by ellipses (...). Entering the ellipses automatically adds one of the switch's 6-byte MAC addresses in the ESI portion and 0 in the selector portion of the address.

Caution ATM addressing can lead to conflicts if you do not configure it correctly. For example,when configuring a new ATM address, you must remove the old one from the configuration.

When the switch initially powers on without previous configuration data, the ATM interfaces configure automatically on the physical ports. The DSLAM uses ILMI and the physical card type to automatically derive:

- ATM interface type
- UNI version
- Maximum virtual path identifier (VPI) and virtual channel identifier (VCI) bits
- ATM interface side
- ATM UNI type

You can accept the default ATM interface configuration or overwrite the default interface configuration using the CLI commands (see the *ATM Switch Router Software Configuration Guide, Chapter 5* Configuring ATM Network Interfaces).

Modifying the Physical Layer Configuration of the Default ATM Interface

This section describes how to modify an ATM interface from the default configuration listed in Chapter 13, "Configuring In-Band Management." You can accept the ATM interface configuration or overwrite the default interface configuration using the CLI commands, which are described in *ATM Switch Router Software Configuration Guide*, *Chapter 6*, Configuring Virtual Connections.

Example

This example describes how to modify an OC-3 interface from the default settings to

- Disable scrambling cell-payload.
- Disable scrambling STS-streaming.
- Change Synchronous Optical Network (SONET) mode of operation from Synchronous Time Stamp level 3c (STS-3c) mode to Synchronous Transfer Module level 1 (STM-1).

To change the configuration of an ATM interface, follow these steps:

Step	Command	Task
1.	DSLAM# **configure terminal**	Enter global configuration mode.
2.	DSLAM(config)# **interface atm** *slot/port*	Select the physical interface to be configured and enter interface configuration mode.
3.	DSLAM(config-if)# **no scrambling cell-payload**	Disable cell-payload scrambling.
4.	DSLAM(config-if)# **no scrambling sts-stream**	Disable STS-stream scrambling.
5.	DSLAM(config-if)# **sonet** {**stm-1** | **sts-3c**}	Configure SONET mode as SDH/STM-1.
6.	DSLAM(config-if)# **end**	Return to priviledged EXEC mode.
7.	DSLAM#	

Example

This example shows how to disable cell-payload scrambling and STS-stream scrambling and changes the SONET mode of operation to Synchronous Digital Hierarchy/Synchronous Transfer Module 1 (SDH/STM-1) of OC-3 physical interface 0/0:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# no scrambling cell-payload
DSLAM(config-if)# no scrambling sts-stream
DSLAM(config-if)# sonet stm-1
DSLAM(config-if)# exit
DSLAM(config)#
```

To display the physical interface configuration, use these privileged EXEC commands:

Command	Task
show controller atm *slot/port*	Show the physical layer configuration.
show running-config	Show the physical layer scrambling configuration.

Examples

This example displays the OC-3 physical interface configuration after you modify the defaults:

```
DSLAM# show controller atm 0/0
Interface ATM0/0 is up
  Hardware is IDT252
 PCI configuration registers:
  bus_no=0, device_no=1
  DeviceID=0x0004, VendorID=0x111D, Command=0x0006, Status=0x0290
  Class=0x02/0x03/0x00, Revision=0x01, LatencyTimer=0x20, CacheLineSize=0x04
  BaseAddr0=0x00000001, BaseAddr1=0x12001000, MaxLat=0x05, MinGnt=0x05
  SubsysDeviceID=0x0000, SubsysVendorID=0x0000

  slot 0, unit 0, subunit 0, fci_type 0x00000001,
                max_pak_size 4528
  particle size 576, pool size 400, cache size 1024, cache end 513
NICStAR registers:
```

```
data[0]: 76
config: 32A19838
status: F00404
rxStatQH: 3C17390
cellDropCt: 0
vpiVciLookupErrorCt: 0
invalidCellCt: 0
rawCellHead: 3CA7440
rawCellHandle: 3CDE004
timer: 7EAAE9
tstBase: 40000
txStatQB: 3C12000
txStatQH: 0
txStatQT: 3C12BC8
genPurpose: 8002
vpiVciMsbMask: 0
abrVbrSchTableDesc: 104C000
abrReadyQueuePtr: 0
vbrReadyQueuePtr: 0
rateTableDesc: 14000
txConnState: 70800068
currentTxSchAddr: 403D4
freeBufQueue0Sz: E000000A
freeBufQueue0Sz: E000000A
freeBufQueue1Sz: E000000B
RECEIVE CONNECTION TABLE:
VCD    Control   Buffer Handle   DMA Address
35  E02A8000  0  0
36  E02A8000  0  0
37  E02A8000  0  0
38  E02A8000  0  0
39  E02A8000  0  0
40  E02A8000  0  0
41  E02A8000  0  0
42  E02A8000  0  0
43  FD2A8000  77  3C97F40
44  FD2A8000  FF  3CD10C0
45  E02A8000  0  0
46  E02A8000  0  0
47  E02A8000  0  0
48  E02A8000  0  0
49  E02A8000  0  0
50  E02A8000  0  0
51  E02A8000  0  0
52  E02A8000  0  0
53  E02A8000  0  0
54  E02A8000  0  0
55  FD2A8000  1B2  3CB7710
56  FD2A8000  176  3CC11F0
57  E02A8000  0  0
58  E02A8000  0  0
59  FD2A8000  1B5  3CB6F90
60  FD2A8000  194  3CA5BF0

  enabled 0, disabled 0, throttled 0
  vc_per_vp 4096, max_vp 1, max_vc 4096, total_vc 9594
Device values:
  IDT252    device number 0, base addr 0xB2001000,
           pci base off 0xA0DEAD01
 TX Status Queue Base 0xA3C12000
 TX Status Queue Tail 0xBF8
 Segmentation Channel Queue 0xA3C14000
 Rcv Stat Queue 0xA3C16000
 Rcv Stat Queue tail A3C17490
```

```
 FreeBufQ0Count 0 FreeBufQ0H 0 FreeBufQ0T 0
 FreeBufQ1Count 2 FreeBufQ1H A3C18510 FreeBufQ1T A3C18BD0
 Free Buff Queue 0 0xA3C18000
 Free Buff Queue 1 0xA3C18100
 Tx Buff Queue 0xA3C1A100
```

This example displays the OC-3 physical layer scrambling configuration after you modify the defaults:

```
DSLAM# show running-config
Building configuration...

Current configuration : 12235 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname DSLAM
!
boot system flash:ni2-dsl-mz.v121_7_da.20010416
slot 1 ATUC-1-4DMT
slot 2 ATUC-1-4DMT
slot 3 ATUC-1-4DMT
slot 4 ATUC-1-4DMT
slot 5 ATUC-1-4DMT
slot 6 ATUC-1-4DMT
slot 7 ATUC-1-4DMT
slot 8 ATUC-1-4DMT
slot 9 ATUC-4FLEXIDMT
slot 10 NI-2-DS3-T1E1
slot 12 ATUC-1-4DMT
slot 13 ATUC-4FLEXIDMT
slot 14 STUC-4-2B1Q-DIR-1
slot 15 STUC-4-2B1Q-DIR-1
slot 16 STUC-4-2B1Q-DIR-1
slot 17 STUC-4-2B1Q-DIR-1
slot 18 ATUC-1-DMT8
slot 19 ATUC-1-4DMT
slot 20 ATUC-1-DMT8
slot 21 ATUC-1-4DMT
slot 22 ATUC-1-4DMT
slot 23 ATUC-1-4DMT
slot 24 ATUC-1-4DMT
slot 25 ATUC-1-4DMT
slot 26 ATUC-1-4DMT
slot 27 ATUC-4FLEXIDMT
slot 28 ATUC-1-4DMT
slot 29 ATUC-1-DMT8
slot 30 ATUC-1-4DMT
slot 31 STUC-4-2B1Q-DIR-1
slot 32 ATUC-1-4DMT-I
no logging console
enable password cisco
!
!
!
!
!
!
dsl-profile default
 alarms
```

```
 dmt check-bytes interleaved downstream 4 upstream 6
 dmt codeword-size downstream 16 upstream 8
 sdsl bitrate 528
!
!
atm oam max-limit 1600
no atm oam intercept end-to-end
atm address 47.0091.8100.0000.0001.64ff.a980.0001.64ff.a980.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
atm ni2-switch trunk ATM0/IMA0
!
icm size 4194304
!
!
interface ATM0/0
 no ip address
 atm maxvp-number 0
 atm maxvc-number 4096
 atm maxvpi-bits 4
!
interface Ethernet0/0
 ip address 172.21.186.145 255.255.255.192
!
interface ATM0/2
 no ip address
 no atm ilmi-keepalive
 atm oam 0 5 seg-loopback
 atm oam 0 16 seg-loopback
 clock source loop-timed
 framing crc4
 lbo short gain10
 ima-group 0
!
ip default-gateway 172.21.186.129
ip classless
ip route 0.0.0.0 0.0.0.0 172.21.186.129
no ip http server
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password cisco
 login
!
end
```

Configuring IP Interface Parameters

This section describes how to configure IP addresses on the DSLAM processor interfaces. You configure each IP address for one of the following types of connections:

- Ethernet port—Configure either from the BOOTP server or by using the **ip address** command in interface-configuration mode for the Ethernet 0/0 interface.

- Serial Line Internet Protocol/Point-to-Point Protocol (SLIP/PPP)—See Chapter 2, "Configuring Terminal Lines and Modem Support."

Note These IP connections are used only for network management.

To configure the DSLAM to communicate using the Ethernet interface, provide the IP address and subnet mask bits for the interface as described in this section.

Defining an IP address

This section provides a summary of IP addressing concepts for those who are familiar with IP addressing.

Internet addresses are 32-bit values assigned to hosts that use the IP protocols. These addresses are in dotted decimal format (four decimal numbers separated by periods), such as 192.17.5.100. Each number is an 8-bit value between 0 and 255.

IP addresses are divided into three classes. These classes differ in the number of bits allocated to the *network* and *host* portions of the address:

- The Class A Internet address format allocates the highest 8 bits to the network field and sets the highest-order bit to 0 (zero). The remaining 24 bits form the host field.
- The Class B Internet address allocates the highest 16 bits to the network field and sets the two highest-order bits to 1, 0. The remaining 16 bits form the host field.
- The Class C Internet address allocates the highest 24 bits to the network field and sets the three highest-order bits to 1, 1, 0. The remaining 8 bits form the host field.

The default IP address is none.

Enter your Internet address in dotted decimal format for each interface you plan to configure.

Defining Subnet Mask Bits

Subnetting is an extension of the Internet addressing scheme which allows multiple physical networks to exist within a single Class A, B, or C network. The subnet mask determines whether subnetting is in effect on a network. The usual practice is to use a few of the far-left bits in the host portion of the network address to assign a subnet field.

Internet addressing conventions allow a total of 24 host bits for Class A addresses, 16 host bits for Class B addresses, and 8 host bits for Class C addresses. When you are further subdividing your network (that is, subnetting your network), the number of host addressing bits is divided between subnetting bits and actual host address bits. You must specify a minimum of two host address bits, or the subnetwork is not populated by hosts.

Note Because all zeros in the host field specifies the entire network, subnetting with subnet address 0 is illegal and is strongly discouraged.

Table 3-2 provides a summary of subnetting parameters.

Table 3-2 Subnetting Parameters

First Class	First Byte	Network Bits	Host Bits	
			Max Subnet Bits	Min Address Bits
A	1 to 126	8	22	2
B	128 to 191	16	14	2

You define subnet mask bits as a decimal number between

- 0 and 22 for Class A addresses
- 0 and 14 for Class B addresses
- 0 and 6 for Class C addresses

Note Do not specify 1 as the number of bits for the subnet field. That specification is reserved by Internet conventions.

To configure the IP address, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface ethernet** *slot/port*	Select the interface to be configured.
2.	**ip address** *A.B.C.D sub_net_A.B.C.D*	Configure the IP and subnetwork address.

Example

This example shows how to configure the Ethernet CPU interface 0/0 with IP address 172.20.40.93 and subnetwork mask 255.255.255.0, and displays the interface information:

```
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# interface ethernet 0/0
DSLAM(config-if)# ip address 172.20.40.93 255.255.255.0
DSLAM(config-if)# end
DSLAM# show interface ethernet 0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0001.64ff.a97f (bia 0001.64ff.a97f)
  Internet address is 172.21.186.145/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 4000 bits/sec, 5 packets/sec
  5 minute output rate 2000 bits/sec, 3 packets/sec
     906236 packets input, 202482126 bytes, 0 no buffer
     Received 889038 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
```

```
     0 input packets with dribble condition detected
     163965 packets output, 21172110 bytes, 0 underruns
     0 output errors, 9 collisions, 1 interface resets
     0 babbles, 0 late collision, 33 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Displaying an IP Address

Use the **show running-config** command to display the CPU IP address:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!
version XX.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
boot bootldr bootflash:/tftpboot/rbhide/ls-wp-mz.XXX-X.X.WA4.X.XX
!
ip host-routing
ip rcmd rcp-enable
ip rcmd rsh-enable
ip rcmd remote-username dplatz
ip domain-name cisco.com
ip name-server 198.92.30.32
atm filter-set tod1 index 4 permit time-of-day 0:0 0:0
atm qos default  cbr max-cell-loss-ratio clp1plus0 12
atm qos default  vbr-nrt max-cell-loss-ratio clp1plus0 12
atm address 47.0091.8100.0000.0041.0b0a.1081.0041.0b0a.1081.00
atm address 47.0091.8100.5670.0000.0000.0000.0040.0b0a.1081.00
atm route-optimization percentage-threshold 250
atm router pnni
 node 1 level 56 lowest
  redistribute atm-static
!

<Information Deleted>

!
interface ATM0/1
 no keepalive
!
interface ATM0/0
 no ip address
 no keepalive
 atm maxvp-number 0
 atm pvc 0 any-vci  encap aal5snap
!
interface Ethernet0/0
ip address 172.20.40.93 255.255.255.0
!
no ip classless
atm route 47.0091.8100.0000... ATM0/0 scope 1
atm route 47.0091.8100.0000.00... ATM0/0 e164-address 1234567
```

```
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Testing the Ethernet Connection

After you configure the IP addresses for the Ethernet interface, test for connectivity between the DSLAM and a host. The host can reside anywhere in your network. To test for Ethernet connectivity, use this command in EXEC mode:

Command	Task
ping ip *ip_address*	Test the configuration using the **ping** command. The **ping** command sends an echo request to the host specified in the command line.

For example, to test Ethernet connectivity from the DSLAM to a workstation with an IP address of 172.20.40.201, enter the command **ping ip 172.20.40.201**. If the DSLAM receives a response, this message appears:

```
DSLAM# ping ip 172.20.40.201

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.20.40.201, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1000 ms
```

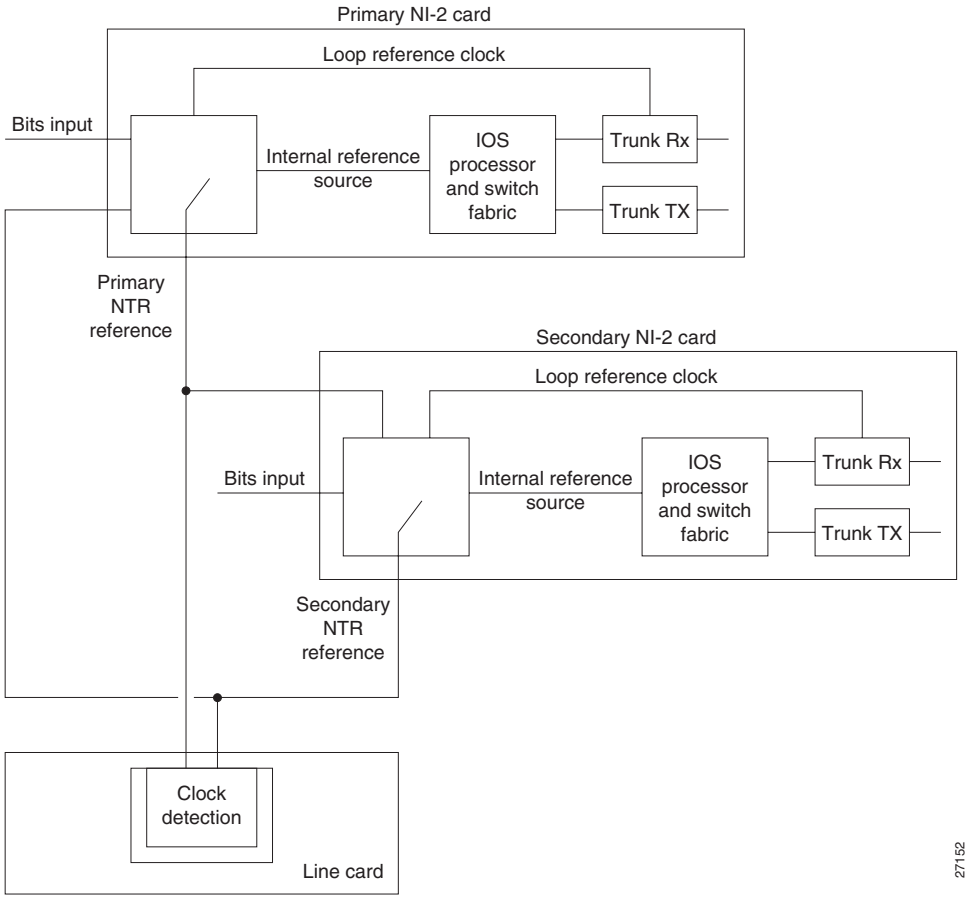
Configuring Network Clocking

This section describes how to configure network clocking and network clocking for the DSLAM. Each port has a transmit clock and derives its receive clock from the receive data. You can configure transmit clocking for each port in one of these ways:

- Network derived—Transmit clocking is derived from the highest priority configured source, either from the internal clock (the default) or the public network.
- Loop-timed—Transmit clocking is derived from the receive clock source.

The DSLAM receives derived clocking, along with data, from a specified interface. For example, in Figure 3-4 the DSLAM extracts transmit clocking, configured as priority one, from the data received at interface 0/1 and distributed as the transmit clock to the rest of the DSLAM. Interface 0/2 then uses network-derived transmit clocking received from interface 0/1.

Figure 3-4 Transmit Clock Distribution



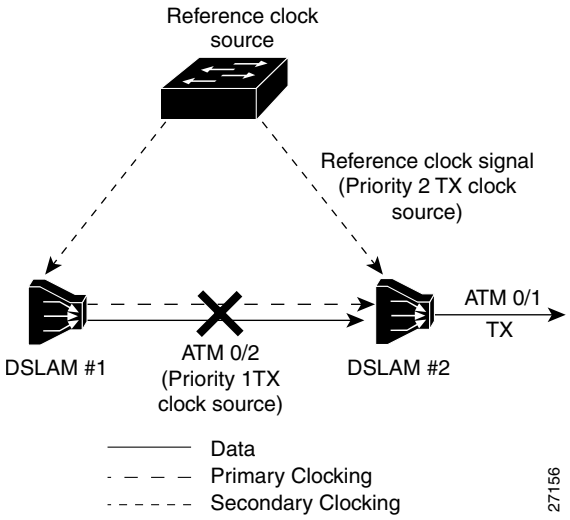
Because the port providing the network clock source could fail, Cisco IOS software provides the ability to configure additional interfaces as clock sources with priorities 1 to 4.

If the network clock source interface stops responding, the software switches to the next highest-configured priority network clock source. For example, Figure 3-5 shows:

- DSLAM number two is configured to receive transmit clocking from an external reference clock source through interface 0/0.
- Interface 0/1 uses network-derived transmit clocking.
- The priority 1 transmit clock interface 0/0 fails.
- The priority 2 interface, 0/2, immediately starts providing the transmit clocking to the backplane and interface 0/1.
- If you configure the **network-clock-select** command as revertive when the priority 1 interface, 0/0, is functioning correctly for at least 1 minute, the interfaces using network-derived transmit clocking starts to receive their clocking again from interface 0/0.

Note The network clock is, by default, configured as non-revertive. Non-revertive means that if a clock fails, the software selects the next-higher clock until that clock fails, then the next-highest, and so forth. The algorithm to switch to the highest priority best clock only runs if you configure the **network-clock-select** command as revertive.

Figure 3-5 Transmit Clocking Priority Configuration Example



These sections describe network clocking:

- Configuring Network Clock Priorities and Sources, page 3-18
- Configuring the Transmit Clocking Source, page 3-19
- Providing Clock Synchronization Services, page 3-22

Configuring Network Clock Priorities and Sources

To configure the network clocking priorities and sources, use these command in global configuration mode:

Command	Task
network-clock-select *priority* {**BITS** | **system** | **atm** *slot/port*}	Configure the priority of a timing source. Priority values are 1 to 4. The trunk, ATM 0/1, is the only ATM interface that can serve as a timing source.
network-clock-select BITS {T1 | E1} *margin*}	Configure the type and margin, in decibels, of the BITS line. Margin values vary according to the length of the T1/E1 line.
network-clock-select revertive	Configure the system to revert to a higher priority timing source when it becomes available.

Examples

This example sets up the DSLAM's building-integrated time source (BITS) interface as the highest-priority clock source, then configures the BITS interface for T1 at 0.6db (0 to 133 feet, or 0 to 40.5 meters).

```
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# network-clock-select 1 BITS
DSLAM(config)# network-clock-select BITS T1 0.6db
```

This example configures interface 0/1, the trunk, as the second-highest priority timing source.

```
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# network-clock-select 2 atm 0/1
```

This example configures the DSLAM's own system clock as the third-highest priority timing source.

```
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# network-clock-select 3 system
```

This example shows how to configure the network clock to revert back to the highest priority clock source after a failure:

```
DSLAM(config)# network-clock-select revertive
DSLAM(config)#
```

Configuring the Transmit Clocking Source

To configure the location from which an interface receives its transmit clocking, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM(config)# **interface atm** *slot/port*	Select the interface to be configured.
2.	DSLAM(config-if)# **clock source** {**loop-timed** | **network-derived**}	Configure the interface network clock source.

Note Network-derived means the highest-priority clock that is both configured and functional.

Examples

This example configures ATM interface 0/1 to receive its transmit clocking from a network-derived source:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# clock source network-derived
DSLAM(config-if)#
```

This example displays the network clocking configuration shown in Figure 4-3:

```
DSLAM# show network-clocks
PLL failed: 58886; PLL Passed: 1082982
FAIL: 0; NCO: F984; REF: F982; ERR: 2; ERR_D: 0; MAG: -1;
clock configuration is NON-Revertive
Priority 1 clock source: BITS clock
Priority 2 clock source: No clock
Priority 3 clock source: No clock
Priority 4 clock source: No clock
Priority 5 clock source: System clock

Current clock source:System clock, priority:5

Nettime Config Register Contents:
NDIV:FF SRC:2, SLOCK:0, TLOCK:0, NFAIL:0, E1:0, NSEL:0
```

```
Trunk LED Register CLK_SEL:3

BITS Register Contents:
CR1: CB, CR2: 0, CR3: 0, ICR: 0, TSR: C1, PSR: 31, ESR: 77, CR4: 0

BITS Source configured as: T1 Short Haul, 0-133ft/0.6db pulse, 100 ohm cable, 1n
```

This example displays the clock source configuration of ATM interface 0/2:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!
version ZZ.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
boot bootldr bootflash:/tftpboot/ls-wp-mz.11X-X.X.WA4.X.XX
!
network-clock-select 2 ATM0/1

<Information Deleted>

!
interface ATM0/2
 no keepalive
 atm manual-well-known-vc
 atm access-group tod1 in
 atm pvc 0 35 rx-cttr 3 tx-cttr 3  interface  ATM0/2 0 any-vci  encap qsaal
 atm route-optimization soft-vc interval 360 time-of-day 18:0 5:0
 clock-source network-derived
!

<Information Deleted>
```

This example displays the interface controller status of interface 0/0:

```
DSLAM# show controllers atm 0/0
Interface ATM0/0 is up
  Hardware is IDT252
 PCI configuration registers:
  bus_no=0, device_no=1
  DeviceID=0x0004, VendorID=0x111D, Command=0x0006, Status=0x0290
  Class=0x02/0x03/0x00, Revision=0x01, LatencyTimer=0x20, CacheLineSize=0x04
  BaseAddr0=0x00000001, BaseAddr1=0x12001000, MaxLat=0x05, MinGnt=0x05
  SubsysDeviceID=0x0000, SubsysVendorID=0x0000

  slot 0, unit 0, subunit 0, fci_type 0x00000001,
                max_pak_size 4528
  particle size 576, pool size 400, cache size 1024, cache end 513
NICStAR registers:
data[0]: 15C
config: 32A19838
status: F00404
rxStatQH: 3C177F0
cellDropCt: 0
vpiVciLookupErrorCt: 0
invalidCellCt: 0
rawCellHead: 3C9F580
```

```
rawCellHandle: 3CDE004
timer: AE396C
tstBase: 40000
txStatQB: 3C12000
txStatQH: 0
txStatQT: 3C13600
genPurpose: 8002
vpiVciMsbMask: 0
abrVbrSchTableDesc: 104C000
abrReadyQueuePtr: 0
vbrReadyQueuePtr: 0
rateTableDesc: 14000
txConnState: 70800002
currentTxSchAddr: 42298
freeBufQueue0Sz: E000000A
freeBufQueue0Sz: E000000A
freeBufQueue1Sz: E000000B
RECEIVE CONNECTION TABLE:
VCD    Control   Buffer Handle   DMA Address
35  E02A8000  0  0
36  E02A8000  0  0
37  E02A8000  0  0
38  E02A8000  0  0
39  E02A8000  0  0
40  E02A8000  0  0
41  E02A8000  0  0
42  E02A8000  0  0
43  FD2A8000  9B  3C98F00
44  FD2A8000  187  3CD0100
45  E02A8000  0  0
46  E02A8000  0  0
47  E02A8000  0  0
48  E02A8000  0  0
49  E02A8000  0  0
50  E02A8000  0  0
51  E02A8000  0  0
52  E02A8000  0  0
53  E02A8000  0  0
54  E02A8000  0  0
55  FD2A8000  1B2  3CB7710
56  FD2A8000  176  3CC11F0
57  E02A8000  0  0
58  E02A8000  0  0
59  FD2A8000  1B5  3CB6F90
60  FD2A8000  194  3CA5BF0

  enabled 0, disabled 0, throttled 0
  vc_per_vp 4096, max_vp 1, max_vc 4096, total_vc 9594
Device values:
  IDT252    device number 0, base addr 0xB2001000,
           pci base off 0xA0DEAD01
 TX Status Queue Base 0xA3C12000
 TX Status Queue Tail 0x1638
 Segmentation Channel Queue 0xA3C14000
 Rcv Stat Queue 0xA3C16000
 Rcv Stat Queue tail A3C178A0
 FreeBufQ0Count 0 FreeBufQ0H 0 FreeBufQ0T 0
 FreeBufQ1Count 1 FreeBufQ1H A3C1A040 FreeBufQ1T A3C1A040
 Free Buff Queue 0 0xA3C18000
 Free Buff Queue 1 0xA3C18100
 Tx Buff Queue 0xA3C1A100
```

Providing Clock Synchronization Services

Any module in a DSLAM chassis capable of receiving and distributing a network timing signal can propagate that signal to any similarly capable module in the chassis. These entities are capable of receiving and distributing a primary reference source (PRS) for the clock:

- A BITS clock through the I/O card
- An OC-3 in a DSLAM chassis
- A quad DS3 module in a DSLAM chassis that derives the clock from the trunk interface

Note A trunk port can propagate a clocking signal in either direction.

If you issue the **network-clock-select** command with the appropriate parameters, you can define a particular port in a DSLAM chassis (subject to the above limitations) to serve as the source of a PRS for the entire chassis or for other devices in the networking environment. This command is described in the "Configuring Network Clock Priorities and Sources" section on page 3-18.

You can also use the **network-clock-select** command to designate a particular port in a DSLAM chassis to serve as a master clock source for distributing a single clocking signal throughout the chassis or to other network devices. You can distribute this reference signal in any location the network needs to globally synchronize the flow of constant bit rate (CBR) data.

Configuring the Network Routing

For network routing, the default software image for the DSLAM contains the PNNI routing protocol. The PNNI protocol provides the route dissemination mechanism for complete plug-and-play capability. This section describes modifications you can make to the default PNNI or Interim-Interswitch Signaling Protocol (IISP) routing configurations.

Use the **atm route** command to configure a static route. Static route configuration allows ATM call setup requests to be forwarded on a specific interface if the addresses match a configured address prefix.

Note An interface must be UNI or IISP if it is configured with a static route. Static routes configured as PNNI interfaces default as down.

Example

This example shows how to use the **atm route** command to configure the 13-byte peer group prefix as 47.0091.8100.567.0000.0ca7.ce01 at interface 0/1:

```
DSLAM(config)# atm route 47.0091.8100.567.0000.0ca7.ce01 atm 0/1
DSLAM(config)#
```

Configuring the Time, Date, and Month

Although not required, you can set several system parameters as part of the initial system configuration. To set the system parameters, perform these tasks, beginning in privileged EXEC mode:

Step	Command	Task
1.	**clock set** *hh:mm:ss day month year*	Set the internal clock.
2.	**configure** [**terminal**]	Enter global configuration mode from the terminal.
3.	**hostname** *name*	Set the system name.

Examples

This example shows how to configure the time, date, and month using the **clock set** command:

```
DSLAM# clock set 15:01:00 17 October 2000
```

This example shows how to configure the host name using the **hostname** command:

```
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# hostname Publications
Publications#
```

This example shows how to confirm the clock setting using the **show clock** command:

```
Publications# show clock
*15:03:12.015 UTC Fri Oct 17 2000
Publications#
```

Configuring SNMP Management

SNMP is an application-layer protocol that allows the SNMP manager and agent to communicate. SNMP provides a message format for sending information between an SNMP manager and an SNMP agent.

The SNMP system consists of three parts:

- SNMP manager
- SNMP agent
- Management Information Bases (MIBs)

The SNMP manager can be part of a network management system (NMS), such as CiscoWorks.

The agent and MIB reside on the DSLAM. To configure SNMP on the DSLAM, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into an agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages that alert the SNMP manager to a condition on the network. Traps can indicate improper user authentication, restarts, link status (up or down), closing of a TCP connection, or loss of connection to a neighbor router or DSLAM.

Figure 3-6 illustrates the communications relationship between the SNMP manager and agent.

Figure 3-6 Communication between an SNMP Agent and Manager

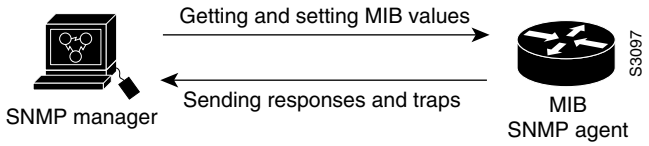


Figure 3-6 shows that a manager can send the agent requests to get and set MIB values. The agent can respond to these requests. Independent of this interaction, the agent can send unsolicited traps to the manager notifying the manager of network conditions.

Cisco supports the SNMP Version 1 protocol, referred to as SNMPv1, and the SNMP Version 2 protocol, referred to as SNMPv2. Cisco's implementation of SNMP supports all MIB II variables (as described in RFC 1213) and SNMP traps (as described in RFC 1215).

RFC 1447, "SNMPv2 Party MIB" (April 1993), describes the managed objects that correspond to the properties associated with SNMPv2 parties, SNMPv2 contexts, and access control policies, as defined by the SNMPv2 Administrative Model. RFC 1450, "SNMPv2 MIB," (April 1993) describes the managed objects that instrument the behavior of an SNMPv2 implementation. Cisco supports the MIB variables as required by the conformance clauses specified in these MIBs.

Cisco provides its own MIB with every system. One of the set of MIB objects provided is the Cisco Entity Asset MIB that enables the SNMP manager to gather data on system card descriptions, serial numbers, hardware and software revision levels, and slot locations.

Although SNMPv2 offers more robust support than SNMPv1, Cisco continues to support SNMPv1. Not all management stations have migrated to SNMPv2, and you must configure the relationship between the agent and the manager to use the version of SNMP supported by the management station.

SNMPv1 offers a community-based form of security defined through an IP address access control list and password. SNMPv2 offers richer security configured through an access policy that defines the relationship between a single manager and agent. SNMPv2 security includes message authentication support using the Message Digest 5 (MD5) algorithm, but because of the Data Encryption Standard (DES) export restrictions, it does not include encryption support through DES. SNMPv2 security provides data origin authentication, ensures data integrity, and protects against message stream modification.

In addition to enhanced security, SNMPv2 support includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism supports the retrieval of tables and large quantities of information, minimizing the number of round-trips required.

The SNMPv2 improved error handling support includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes now report the error type. Three kinds of exceptions are also reported:

- No such object exceptions
- No such instance exceptions
- End of MIB view exceptions

There is no specific command to enable SNMP. The first **snmp-server** command that you enter enables both versions of SNMP.

To configure SNMP support, perform the tasks in the appropriate sections:

- Configuring for both SNMPv1 and SNMPv2
- Configuring SNMPv2 Support

- Configuring SNMPv1 Support

To configure the relationship between the agent and the manager on the DSLAM, you need to know the version of the SNMP protocol that the management station supports. An agent can communicate with multiple managers, so you can configure the Cisco IOS software to support communications with one management station using the SNMPv1 protocol and another using the SNMPv2 protocol.

Configuring Support for Both SNMPv1 and SNMPv2

This section tells you how to configure support for both SNMPv1 and SNMPv2. The topics are:

- Establishing the contact, location, and serial number of the SNMP agent
- Defining the maximum SNMP agent packet size
- Monitoring SNMP status
- Disabling the SNMP agent
- Enabling the SNMP agent shutdown mechanism

Establishing the Contact, Location, and Serial Number of the SNMP Agent

You can set the system contact, location, and serial number of the SNMP agent so that these descriptions can be accessed through the configuration file. Use one or more of these commands in global configuration mode:

Command	Task
snmp-server contact *text*	Set the system contact string.
snmp-server location *text*	Set the system location string.
snmp-server chassis-id *text*	Set the system serial number.

Defining the Maximum SNMP Agent Packet Size

You can set the maximum packet size permitted when the SNMP agent is receiving a request or generating a reply. To do so, use this command in global configuration mode:

Command	Task
snmp-server packetsize *byte-count*	Establish the maximum packet size.

Monitoring SNMP Status

To monitor SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use this command in EXEC mode:

Command	Task
show snmp	Monitor SNMP status.

Disabling the SNMP Agent

To disable both versions of SNMP (SNMPv1 and SNMPv2) concurrently, use this command in global configuration mode:

Command	Task
no snmp-server	Disable SNMP agent operation.

Enabling the SNMP Agent Shutdown Mechanism

This section tells you how to enable a reload from the network after a system shutdown.

Using SNMP packets, a network management tool can send messages to users on both virtual terminals and the console. This facility operates in a similar fashion to the EXEC **send** command, but the SNMP request that causes the message to be issued to the users also specifies the action to be taken after the message is delivered. One possible action is a shutdown request. After a system shuts down, it is typically reloaded.

Reloading from the network is a powerful feature, and therefore is protected by the **snmp-server system-shutdown** global configuration command. If you do not issue this command, the shutdown mechanism does not enable.

To enable the SNMP agent shutdown mechanism, use this command in global configuration mode:

Command	Task
snmp-server system-shutdown	Use the SNMP message reload feature and request a system shutdown message.

To understand how to use this feature with SNMP requests, read the document *mib.txt* available by anonymous FTP from Cisco Connection Online.

Configuring SNMPv2 Support

SNMPv2 security requires that you create an access policy that defines the relationship between a manager and the agent. For each management station that the agent communicates with, you must create a separate access policy. Creating an access policy is a multiple-task process:

Step 1 If you do not want to use one of the predefined views, define a view to identify the objects that can be seen.

Step 2 Define a context to identify the object resources that can be acted on.

Step 3 Define a party for both the manager and the agent to identify them.

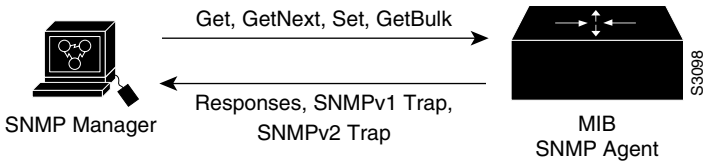
Step 4 Using the definitions created in the previous tasks, configure the access policy that characterizes the communications that can occur between the manager and the agent. The privileges that you define for the access policy depend on whether the agent is defined as the source or the destination. For example:

- When the agent party is defined as the destination in an access policy, the access policy privileges define the management operations that the agent will accept from the manager and perform in relation to the object resources.

- When the agent party is defined as the source in an access policy, the access policy privileges define the responses and traps that the agent can send to the manager.

Figure 3-7 shows the information exchanged between the manager and the agent.

Figure 3-7 Flow of Management Operations Requests, Responses, and Traps



- The top arrow, leading from the manager to the agent, shows the types of requests the manager can send to the agent.
- The bottom arrow, leading from the agent to the manager, shows the kind of information that the agent can send to the manager.

The agent sends trap messages to the manager in response to certain network conditions. Trap messages are unsolicited and are not related to the request/response communication exchange between the manager and the agent that occurs in relation to MIB variables. For any given manager and agent relationship, the privileges defined in the access policy constrain communications to a specific set of operations.

You must create access policies for

- Each new agent you install
- On an agent when you install new management stations

Each time a network address changes on a management station, you must reconfigure the access policy to reflect the new information for the management station.

Configuring Support for SNMPv2

To configure support for SNMPv2, perform the tasks described in the following sections:

- Creating or Modifying an SNMP View Record
- Creating or Modifying an SNMP Context Record
- Creating or Modifying an SNMPv2 User Record
- Creating an SNMPv2 Access Policy
- Defining SNMPv2 Trap Operations

After you create a record, you can modify the record contents by changing one or more of the record values. To do this, issue the command again, naming the record that you created originally. You must fully specify the record values, including the argument values, to remain unchanged.

Creating or Modifying an SNMP View Record

To create or modify an SNMP view record, use this command in global configuration mode:

Command	Task
snmp-server view *view-name oid-tree* {**included** | **excluded**}	Create or modify a view record.

To remove a view record, use the **no snmp-server view** command.

Creating or Modifying an SNMP Context Record

To create or modify an SNMP context record, use this command in global configuration mode:

Command	Task
snmp-server context *context-name context-oid view-name*	Create or modify a context record.

To remove a context entry, use the **no snmp-server context** command. Specify only the name of the context. The name identifies the context to be deleted.

Creating or Modifying an SNMPv2 User Record

To create or modify an SNMPv2 user record, use this command in global configuration mode:

Command	Task
snmp-server user *user-name user-oid* [*protocol-address*] [**packetsize** *size*] [**local** | **remote**] [**authentication md5** *key* [**clock** *clock*] [**lifetime** *lifetime*]	Create or modify a user record.

To remove a user record, use the **no snmp-server user** command.

Creating an SNMPv2 Access Policy

To create or modify an SNMPv2 access policy, use this command in global configuration mode:

Command	Task
snmp-server access-policy *destination-party source-party context privileges*	Create or modify an access policy.

To remove an SNMPv2 access policy, use the **no snmp-server access-policy** command. Specify all three arguments to correctly identify the access policy to be deleted. A difference of one value constitutes a unique access policy entry.

Defining SNMPv2 Trap Operations

A trap is an unsolicited message sent by an SNMP agent to an SNMP manager indicating that some event has occurred. The SNMP trap operations allow you to configure the Cisco IOS software to send information to a network management application when a particular event occurs. You can specify these features for SNMPv2 agent trap operations:

- Source interface
- Recipient of the trap message
- Trap message authentication
- Trap types
- Retransmission interval
- Message (packet) queue length for each trap host

To define the recipient of the trap message, configure a user record for the manager, including the protocol address, and specify the user record as the destination user for the **snmp-server access policy** command.

To define traps for the agent to send to the manager, use one or more of these commands in global configuration mode:

Command	Task
snmp-server trap-source *interface*	Specify the source interface (and hence IP address) of the trap message.
snmp-server access-policy *destination-user source-user context privileges*	Specify the access policy that defines the traps that the agent can send to the manager.
snmp-server enable traps snmp authentication [*snmpv1* | *snmpv2*]	Establish trap message authentication.
snmp-server trap-timeout *seconds*	Define how often to resend trap messages on the retransmission queue.
snmp-server queue-length *length*	Establish the message queue length for each trap host.

Because SNMP traps are inherently unreliable but too important to lose, the DSLAM stores at least one syslog message (the most recent trap), in a history table. You can specify the level of syslog traps (Cisco Syslog MIB) stored in the history table and sent to the SNMP network management station.

Configuring SNMPv1 Support

If the SNMP manager supports only the SNMPv1 protocol, you must configure the relationship between the manager and the agent using SNMPv1 support.

Using the **snmp-server community** command, specify a string and, optionally, a MIB view and an access list. The string is used as a password. The MIB view defines the subset of all MIB objects that the given community can access. The access list identifies the IP addresses of systems on which SNMPv1 managers reside that might use the community string to gain access to the SNMPv1 agent.

To configure support for SNMPv1, perform the tasks in these sections:

Creating or Modifying Access Control for an SNMPv1 Community

You can configure a community string, which acts like a password, to permit access to the agent on the DSLAM. Optionally, you can associate a list of IP addresses with that community string to permit only managers with these IP addresses to use the string.

To configure a community string, use this command in global configuration mode:

Command	Task
snmp-server community *string* [**view** *view-name*] [**ro** | **rw**] [*access-list number*]	Define the community access string.

You can configure one or more community strings. To remove a specific community string, use the **no snmp-server community** command.

Defining SNMP Trap Operations for SNMPv1

The SNMP trap operations allow a system administrator to configure the agent to send information to a manager when a particular event occurs. You can specify these features for SNMP server trap operations:

- Source interface
- Recipient
- Trap message authentication
- Trap types
- Retransmission interval
- Define the message (packet) queue length for each trap host

To define traps for the agent to send to the specified manager, perform these tasks in global configuration mode:

Step	Command	Task
1.	**snmp-server trap-source** *interface*	Specify the source interface (and IP address) of the trap message.
2.	**snmp-server host** *address community-string* [*trap-type*]	Specify the recipient of the trap message.
3.	**snmp-server enable traps snmp authentication**	Establish trap message authentication.
4.	**snmp-server trap-timeout** *seconds*	Define how often to resend trap messages on the retransmission queue.
5.	**snmp-server queue-length** *length*	Establish the message queue length for each trap host.

Because SNMP traps are inherently unreliable but too important to lose, at least one syslog message, the DSLAM stores the most recent trap in a history table. You can specify the level of syslog traps (Cisco Syslog MIB) stored in the history table and sent to the SNMP network management station.

Configuring SNMP RMON Support

Remote Monitoring (RMON)

- Provides visibility of individual nodal activity
- Allows you to monitor all nodes and their interaction on a LAN segment

RMON, used in conjunction with the SNMP agent in the DSLAM, allows you to

- View traffic that flows through the DSLAM
- View segment traffic not necessarily destined for the DSLAM

Combining RMON alarms and events with existing MIBs allows you to choose where monitoring occurs.

RMON can be very data- and processor-intensive. Measure usage effects to ensure that DSLAM performance is not degraded and to minimize excessive management traffic overhead. Native mode is less intensive than promiscuous mode.

Cisco IOS software images are available in versions with or without the explicit RMON option. Images without the explicit RMON option include limited RMON support (RMON alarms and event groups only). Images with the RMON option include support for all nine groups (statistics, history, alarms, hosts, hostTopN, matrix, filter, capture, and event). As a security precaution, support for the packet capture group allows capture of packet header information only; data payloads are not captured.

Note This section describes general SNMP RMON configuration. See the *ATM Switch Router Software Configuration Guide, Chapter 14*, Configuring ATM Accounting and ATM RMON for ATM RMON configuration.

To set an RMON alarm or event, use one of these commands in global configuration mode:

Command	Task
rmon alarm *number variable interval* {**delta** | **absolute**} **rising-threshold** *value* [*event-number*] **falling-threshold** *value* [*event-number*] [**owner** *string*]	Set an alarm on a MIB object.
rmon event *number* [**log**] [**trap** *community*] [**description** *string*] [**owner** *string*]	Add or remove an event in the RMON event table.

You can set an alarm on any MIB object in the access server. To disable an alarm, you must enable the **no** form of this command on each alarm you configure. You cannot disable all the alarms you configure at one time.

Refer to RFC 1757 to learn more about alarms and events and how they interact with each other.

To display the current RMON status, use these EXEC commands:

Command	Task
show rmon or **show rmon task**	Display general RMON statistics.
show rmon alarms	Display the RMON alarm table.
show rmon events	Display the RMON event table.

Examples

This example shows how to enable the **rmon event** command:

```
DSLAM# rmon event 1 log trap eventtrap description "High ifOutErrors" owner sdurham
```

This example shows how to configure this RMON event:

- RMON event number 1
- Defined as *High ifOutErrors*
- Generates a log entry when the event is triggered by an alarm
- User *sdurham* owns the row that is created in the event table by this command
- Generates a SNMP trap when the event is triggered

This example shows how to configure an RMON alarm using the **rmon alarm** command:

```
DSLAM# rmon alarm 10 ifEntry.20.1 20 delta rising-threshold 15 1 falling-threshold 0 owner
jjjohnson
```

This example shows how to configure this RMON alarm:

- RMON alarm number 10.
- The alarm monitors the MIB variable *ifEntry.20.1* one time every 20 seconds until the alarm is disabled and checks the change in the rise or fall of the variable.
- If the *ifEntry.20.1* value shows a MIB counter increase of 15 or more, such as from 100000 to 100015, the alarm is triggered.
- The alarm in turn triggers event number 1, which is configured using the **rmon event** command.

Possible events include a log entry or an SNMP trap. If the *ifEntry.20.1* value changes by 0, the alarm is reset and can be triggered again.

Storing the Configuration

After you complete autoconfiguration and any manual configurations, copy the configuration into nonvolatile random-access memory (NVRAM). If you power off your DSLAM prior to saving the configuration in NVRAM you lose all manual configuration changes.

An example of the **copy running-config** command is:

```
DSLAM# copy running-config startup-config
Building configuration...
[OK]
```

Testing the Configuration

After you finish configuring the DSLAM, you can use the commands described in this section to confirm the hardware, software, and interface configuration:

- Confirming the Hardware Configuration
- Confirming the Software Version
- Confirming Power-on Diagnostics
- Confirming the Ethernet Configuration
- Confirming the ATM Address
- Testing the Ethernet Connection
- Confirming the ATM Connections
- Confirming the ATM Interface Configuration
- Confirming the Interface Status
- Confirming Virtual Channel Connections
- Confirming the Running Configuration
- Confirming the Saved Configuration

Confirming the Hardware Configuration

Use the **show hardware** command to confirm the correct hardware installation. For example:

```
DSLAM# show hardware
Chassis Type: C6260
I/O Card: 6260-E1-IO

Slot 1 : ATUC-1-4DMT                        Slot 17: STUC-4-2B1Q-DIR-1
Slot 2 : ATUC-1-4DMT                        Slot 18: ATUC-1-DMT8
Slot 3 : ATUC-1-4DMT                        Slot 19: ATUC-1-4DMT
Slot 4 : ATUC-1-4DMT                        Slot 20: ATUC-1-DMT8
Slot 5 : ATUC-1-4DMT                        Slot 21: ATUC-1-4DMT
Slot 6 : ATUC-1-4DMT                        Slot 22: ATUC-1-4DMT
Slot 7 : ATUC-1-4DMT                        Slot 23: ATUC-1-4DMT
Slot 8 : ATUC-1-4DMT                        Slot 24: ATUC-1-4DMT
Slot 9 : ATUC-4FLEXIDMT                     Slot 25: ATUC-1-4DMT
Slot 10: NI-2-DS3-T1E1                      Slot 26: ATUC-1-4DMT
Slot 11: EMPTY                              Slot 27: ATUC-4FLEXIDMT
Slot 12: ATUC-1-4DMT                        Slot 28: ATUC-1-4DMT
Slot 13: ATUC-4FLEXIDMT                     Slot 29: ATUC-1-DMT8
Slot 14: STUC-4-2B1Q-DIR-1                  Slot 30: ATUC-1-4DMT
Slot 15: STUC-4-2B1Q-DIR-1                  Slot 31: STUC-4-2B1Q-DIR-1
Slot 16: STUC-4-2B1Q-DIR-1                  Slot 32: ATUC-1-4DMT-I

Fan Module 1: Present   2: Present

Power Supply Module 1: 6260-PEM-AC
Power Supply Module 2: 6260-PEM-AC
```

Confirming the Software Version

Use the **show version** command to confirm the correct version and type of software and the configuration register are installed. For example:

```
DSLAM# show version
Cisco Internetwork Operating System Software
IOS (tm) NI2 Software (NI2-DSL-M), Experimental Version 12.1(20010416:212622) []
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Mon 16-Apr-01 17:26 by chrel
Image text-base: 0x800082C0, data-base: 0x8132A000

ROM: System Bootstrap, Version 12.0(5)DA, EARLY DEPLOYMENT RELEASE SOFTWARE (fc)
BOOTFLASH: NI2 Software (NI2-DBOOT-M), Version 12.1(6)DA, EARLY DEPLOYMENT RELE)

6260_E1IMA uptime is 1 week, 6 days, 5 hours, 48 minutes
System returned to ROM by reload
System image file is "flash:ni2-dsl-mz.v121_7_da.20010416"

cisco 6260 (NI2) processor with 60416K/5120K bytes of memory.
RC64475 CPU at 100Mhz, Implementation 48, Rev 0.0
Bridging software.
1 Ethernet/IEEE 802.3 interface(s)
112 DMT DSL Port interface(s)
20 SDSL DSL Port interface(s)
13 ATM network interface(s)
522232 bytes of non-volatile configuration memory.

4096K bytes of Boot Flash (Sector size 128K).
16384K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x2102
```

Confirming the Ethernet Configuration

Use the **show interface ethernet** command to confirm the Ethernet interface is configured correctly. For example:

```
DSLAM# show interface ethernet 0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0001.64ff.a97f (bia 0001.64ff.a97f)
  Internet address is 172.21.186.145/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 2000 bits/sec, 3 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
     910869 packets input, 202979554 bytes, 0 no buffer
     Received 890807 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     166029 packets output, 21332341 bytes, 0 underruns
     0 output errors, 9 collisions, 1 interface resets
     0 babbles, 0 late collision, 33 deferred
```

```
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Confirming the ATM Address

Use the **show atm addresses** command to confirm correct configuration of the ATM address for the DSLAM. For example:

```
DSLAM# show atm addresses

Switch Address(es):
  47.0091.8100.0000.0001.64ff.a980.0001.64ff.a980.00 active
  NOTE: Switch addresses with selector bytes 01 through 7F
        are reserved for use by PNNI routing

PNNI Local Node Address(es):
  47.0091.8100.0000.0001.64ff.a980.0001.64ff.a980.01 Node 1

Soft VC Address(es):
  47.0091.8100.0000.0001.64ff.a980.4000.0c98.0020.00 ATM0/2
  47.0091.8100.0000.0001.64ff.a980.4000.0c98.0030.00 ATM0/3

Soft VC Address(es) for Frame Relay Interfaces :

ILMI Switch Prefix(es):
  47.0091.8100.0000.0001.64ff.a980

ILMI Configured Interface Prefix(es):

LECS Address(es):
```

Testing the Ethernet Connection

After you configure the IP addresses for the Ethernet interface, test for connectivity between the DSLAM and a host. The host can reside in any location on your network. To test for Ethernet connectivity, use this command:

Command	Task
ping ip *ip_address*	Test the configuration using the **ping** command. The **ping** command sends an echo request to the host specified in the command line.

For example, to test Ethernet connectivity from the DSLAM to a workstation with an IP address of 172.20.40.201, enter the command **ping ip 172.20.40.201**. If the DSLAM receives a response, this message appears:

```
DSLAM# ping ip 172.20.40.201

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.20.40.201, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1000 ms
```

Confirming the ATM Connections

Use the **ping atm** command to confirm that the ATM interfaces are configured correctly. For example:

```
DSLAM# ping atm interface atm 0/1 5 seg-loopback

Type escape sequence to abort.
Sending Seg-Loopback 5, 53-byte OAM Echoes to a neighbour,timeout is 5 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
DSLAM#
```

Confirming the ATM Interface Configuration

Use the **show atm interface** command to confirm the atm interfaces are configured correctly. For example:

```
DSLAM# show atm interface

Interface:      ATM0/0          Port-type:      cpu
IF Status:      UP              Admin Status:   UP
Auto-config:    disabled        AutoCfgState:   not applicable
IF-Side:        not applicable  IF-type:        not applicable
Uni-type:       not applicable  Uni-version:    not applicable
Max-VPI-bits:   4               Max-VCI-bits:   14
Max-VP:         0               Max-VC:         4096
ConfMaxSvpcVpi: 0               CurrMaxSvpcVpi: 0
ConfMaxSvccVpi: 0               CurrMaxSvccVpi: 0
ConfMinSvccVci: 35              CurrMinSvccVci: 35
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   TVCLs   PVPLs SoftVPLs   SVPLs Total-Cfgd Inst-Conns
     26        0       0       0       0        0       0         26         26
Logical ports(VP-tunnels):     0
Input cells:    106840          Output cells:   0
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 0, Output AAL5 pkts: 0, AAL5 crc errors: 0


Interface:      ATM0/2          Port-type:      e1_ima_link
IF Status:      UP              Admin Status:   UP
Auto-config:    enabled         AutoCfgState:   waiting for response from peer
IF-Side:        Network         IF-type:        UNI
Uni-type:       Private         Uni-version:    V3.0
Max-VPI-bits:   8               Max-VCI-bits:   14
Max-VP:         255             Max-VC:         16383
ConfMaxSvpcVpi: 255             CurrMaxSvpcVpi: 255
ConfMaxSvccVpi: 255             CurrMaxSvccVpi: 255
ConfMinSvccVci: 35              CurrMinSvccVci: 35
Svc Upc Intent: pass            Signalling:     Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0001.64ff.a980.4000.0c98.0020.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   TVCLs   PVPLs SoftVPLs   SVPLs Total-Cfgd Inst-Conns
      2        0       0       0       1        0       0         3          2
Logical ports(VP-tunnels):     0
Input cells:    925             Output cells:   74
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 0, Output AAL5 pkts: 0, AAL5 crc errors: 0

          [additional interfaces deleted]
```

Confirming the Interface Status

Use the **show atm status** command to confirm the status of ATM interfaces. For example:

```
DSLAM# show atm status
NUMBER OF INSTALLED CONNECTIONS: (P2P=Point to Point, P2MP=Point to MultiPoint,)

Type     PVCs SoftPVCs     SVCs     TVCs     PVPs SoftPVPs     SVPs      Total
P2P        26        0        0        0        0        0        0         26
P2MP        0        0        0        0        0        0        0          0
MP2P        0        0        0        0        0        0        0          0
                                      TOTAL INSTALLED CONNECTIONS =         26

PER-INTERFACE STATUS SUMMARY AT 07:15:04 UTC Wed Oct 18 2000:
   Interface      IF         Admin  Auto-Cfg    ILMI Addr     SSCOP    Hello
     Name       Status      Status    Status    Reg State     State    State
------------- -------- ------------ -------- ------------ --------- --------
ATM0/0             UP           up      n/a  UpAndNormal      Idle      n/a
ATM0/2             UP           up  waiting   Restarting      Idle      n/a
```

Confirming Virtual Channel Connections

Use the **show atm vc** command to confirm the status of ATM virtual channels. For example:

```
DSLAM# show atm vc
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM0/0       0     36     PVC     ATM0/2       0     16    ILMI   DOWN
ATM0/0       0     38     PVC     ATM0/2       0     5     QSAAL  DOWN
ATM0/0       0     500    PVC     ATM0/1       0     500   SNAP   UP
ATM0/1       0     500    PVC     ATM0/0       0     500   SNAP   UP
ATM0/2       0     5      PVC     ATM0/0       0     38    QSAAL  DOWN
ATM0/2       0     16     PVC     ATM0/0       0     36    ILMI   DOWN
```

Confirming the Running Configuration

Use the **show running-config** command to confirm that the configuration being used is configured correctly. For example:

```
DSLAM# show running-config
Building configuration...

Current configuration : 12407 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 6260_E1IMA
!
boot system flash:ni2-dsl-mz.v121_7_da.20010416
slot 1 ATUC-1-4DMT
slot 2 ATUC-1-4DMT
slot 3 ATUC-1-4DMT
slot 4 ATUC-1-4DMT
slot 5 ATUC-1-4DMT
```

```
slot 6 ATUC-1-4DMT
slot 7 ATUC-1-4DMT
slot 8 ATUC-1-4DMT
slot 9 ATUC-4FLEXIDMT
slot 10 NI-2-DS3-T1E1
slot 12 ATUC-1-4DMT
slot 13 ATUC-4FLEXIDMT
slot 14 STUC-4-2B1Q-DIR-1
slot 15 STUC-4-2B1Q-DIR-1
slot 16 STUC-4-2B1Q-DIR-1
slot 17 STUC-4-2B1Q-DIR-1
slot 18 ATUC-1-DMT8
slot 19 ATUC-1-4DMT
slot 20 ATUC-1-DMT8
slot 21 ATUC-1-4DMT
slot 22 ATUC-1-4DMT
slot 23 ATUC-1-4DMT
slot 24 ATUC-1-4DMT
slot 25 ATUC-1-4DMT
slot 26 ATUC-1-4DMT
slot 27 ATUC-4FLEXIDMT
slot 28 ATUC-1-4DMT
slot 29 ATUC-1-DMT8
slot 30 ATUC-1-4DMT
slot 31 STUC-4-2B1Q-DIR-1
slot 32 ATUC-1-4DMT-I
no logging console
enable password cisco
!
!
!
!
!
!
dsl-profile default
 alarms
 dmt check-bytes interleaved downstream 4 upstream 6
 dmt codeword-size downstream 16 upstream 8
 sdsl bitrate 528
!
atm oam max-limit 1600
no atm oam intercept end-to-end
atm address 47.0091.8100.0000.0001.64ff.a980.0001.64ff.a980.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
atm ni2-switch trunk ATM0/IMA0
!
icm size 4194304
!
!
interface ATM0/0
 no ip address
 atm maxvp-number 0
 atm maxvc-number 4096
 atm maxvpi-bits 4
!
interface Ethernet0/0
 ip address 172.21.186.145 255.255.255.0
!
interface ATM0/2
 no ip address
```

```
 no atm ilmi-keepalive
 atm oam 0 5 seg-loopback
 atm oam 0 16 seg-loopback
 clock source loop-timed
 framing crc4
 lbo short gain10
 ima-group 0
!
ip default-gateway 172.21.186.129
ip classless
ip route 0.0.0.0 0.0.0.0 172.21.186.129
no ip http server
!
atm route 47.0091.8100.5670.0000.ca7c.e01... ATM0/0
snmp-server trap-source ATM0/0
snmp-server enable traps config
snmp-server enable traps alarms
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password cisco
 login
!
end
```

Confirming the Saved Configuration

Use the **show startup-config** command to confirm that the configuration saved in NVRAM is configured correctly. For example:

```
DSLAM# show startup-config
Using 1657 out of 522232 bytes
!
! Last configuration change at 11:35:31 EDT Thu Jun 3 1999
! NVRAM config last updated at 11:40:08 EDT Thu Jun 3 1999
!
version XX.X
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname ni2-3
!
enable password lab
!
!
dmt-profile default
network-clock-select 1 ATM0/1
network-clock-select 2 system
ip subnet-zero
ip host-routing
ip domain-name cisco.com
ip name-server 171.69.204.11
!
atm address 47.0091.8100.0000.007b.f444.7801.007b.f444.7801.00
atm router pnni
```

```
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
clock timezone EST -5
clock summer-time EDT recurring
!
process-max-time 200
!
interface ATM0/0
 ip address 70.0.0.2 255.0.0.0
 no ip directed-broadcast
 map-group test
 atm cac service-category abr deny
 atm maxvp-number 0
!
interface Ethernet0/0
 ip address 172.27.32.157 255.255.255.0
 no ip directed-broadcast
 no ip proxy-arp
 no keepalive
!
interface ATM0/1
 no ip address
 no ip directed-broadcast
 no atm auto-configuration
 no atm ilmi-keepalive
 no atm address-registration
 no atm ilmi-enable
 atm cac service-category abr deny
 atm manual-well-known-vc
 atm nni
 atm pvc 0 500  interface  ATM0/0 0 500  encap aal5snap
 atm oam 0 500 seg-loopback
!
interface ATM0/2
 no ip address
 no ip directed-broadcast
 no atm ilmi-keepalive
 atm cac service-category abr deny
!
ip default-gateway 172.27.144.4
ip classless
!
!
map-list test
 ip 70.0.0.1 atm-vc 500
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
!
sntp server 171.69.204.139
end
```

CHAPTER 4

Configuring System Management Functions

This chapter describes the basic tasks for configuring Cisco DSLAM general system features such as access control and basic DSLAM management. These sections describe these tasks:

System Management Tasks

The role of the administration interface is to provide a simple, command-line interface to all internal management and debugging DSLAM facilities. This section describes the system management tasks you need to perform to maximize system performance.

Configuring a Command Alias

To create and configure a command alias, perform these tasks in global configuration mode:

Step	Command	Task
1	**alias** *mode alias-name alias-command-line*	Create a command alias.
2	**alias** *mode*	Configure the command mode of the original and alias commands.
3	**alias** *name*	Configure the command alias.

To display all aliases, use the privileged EXEC command:

Command	Task
show aliases [*mode*]	Display all alias commands, or the alias commands in a specified mode.

Configuring Buffers

To make adjustments to initial buffer pool settings and to the limits at which temporary buffers are created and destroyed, use the global configuration command:

Command	Task
buffers {**small** | **middle** | **big** | **large** | **verylarge** | **huge** | *type number*}	Configure buffers. The default buffer size is 18024 bytes.
show buffers [**all** | **alloc** [**dump**]]	Display statistics for the buffer pools on the network server.

To display the buffer pool statistics, use the privileged EXEC command:

Command	Task
show buffers [**all** | **alloc** [**dump**]]	Display statistics for the buffer pools on the network server.

Configuring the Cisco Discovery Protocol

To specify the frequency with which the DSLAM sends Cisco Discover Protocol (CDP) updates, perform the tasks in global configuration mode:

Step	Command	Task
1	**cdp holdtime** *seconds*	Specify the hold time in seconds, to be sent in packets.
2	**cdp timer** *seconds*	Specify the frequency with which your DSLAM sends CDP updates.
3	**cdp run**	Enable CDP.

To reset CDP traffic counters to zero (0) on your DSLAM, perform the tasks in privileged EXEC mode:

Step	Command	Task
1	**clear cdp counters**	Clear CDP counters.
2	**clear cdp table**	Clear CDP tables.

To show the CDP configuration, use the privileged EXEC commands:

Command	Task
show cdp	Display global CDP information.
show cdp entry-name [**protocol** | **version**]	Display information about a neighbor device listed in the CDP table.

Command	Task
show cdp interface [*type number*]	Display interfaces on with CDP enabled.
show cdp neighbors [*interface-type interface-number*] [**detail**]	Display CDP neighbor information.
show cdp traffic	Display CDP traffic information.

Configuring the Enable Password

To log on to the DSLAM at a specified level, use the EXEC command:

Command	Task
enable *level*	Enable login.

To configure the enable password for a given level, use the global configuration command:

Command	Task
enable password [**level** *level*] [*encryption-type*] *password*	Configure the enable password.

Configuring the Load-Interval

To change the length of time for which data is used to compute load statistics, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1	**interface** *type slot/port*	Select the physical interface to be configured.
2	**load-interval** *seconds*	Configure the load interval.

Configuring Logging

To log messages to a syslog server host, use the global configuration commands:

Command	Task
logging *host*	Configure the logging name or IP address of the host to be used as a syslog server.
logging buffered	To log messages to an internal buffer, use the **logging buffered** global configuration command. The **no logging buffered** command cancels the use of the buffer and writes messages to the console terminal, which is the default.

Command (continued)	Task
logging console *level*	To limit messages logged to the console based on severity, use the **logging console** global configuration command.
logging facility *facility-type*	To configure the syslog facility in which error messages are sent, use the **logging facility** global configuration command. To revert to the default of local, use the **no logging facility** global configuration command.
logging monitor *level*	To limit messages logged to the terminal lines (monitors) based on severity, use the **logging monitor** global configuration command. This command limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above the value of *level*. The **no logging monitor** command disables logging to terminal lines other than the console line.
logging on	To control logging of error messages, use the **logging on** global configuration command. This command enables or disables message logging to all destinations except the console terminal. The **no logging on** command enables logging to the console terminal only.
logging synchronous [**level** *severity-level* | **all**] [**limit** *number-of-buffers*]	To synchronize unsolicited messages and **debug** output with solicited DSLAM output and prompts for a specific console port line, auxiliary port line, or virtual terminal line, use the **logging synchronous** line configuration command. Use the **no** form of the command to disable synchronization of unsolicited messages and debug output.
logging trap *level*	To limit messages logged to the syslog servers based on severity, use the **logging trap** global configuration command. The command limits the logging of error messages sent to syslog servers to only those messages at the specified level. The **no logging trap** command disables logging to syslog servers.

Configuring Login Authentication

To enable Extended Terminal Access Controller Access Control System (TACACS+) authentication for logins, perform these steps, beginning in global configuration mode:

Command	Task
line [**aux** | **console** | **vty**] *line-number*	Select the line to configure.
login authentication {**default** | *list-name*}	Configure login authentication.

Configuring the Scheduler

To control the maximum amount of time that can elapse without running the lowest-priority system processes, use these global configuration commands:

Command	Task
scheduler allocate milliseconds *milliseconds*	Configure the scheduler allocate integer that specifies the interval, in milliseconds. The minimum interval that you can specify is 500 milliseconds. There is no maximum value.
scheduler process-watchdog {**hang** | **normal** | **reload** | **terminate**}	Configure **scheduler process-watchdog**.

Configuring Miscellaneous System Services

To configure miscellaneous system services, use these global configuration commands:

Command	Task
service alignment	Configure alignment correction and logging.
service compress-config	Compress the configuration file.
service config	Load config TFTP files.
service decimal-tty	Interpret TTY line numbers in decimal.
service exec-callback	Enable EXEC callback.
service exec-wait	Configure a delay of the startup of the EXEC on noisy lines.
service finger	Allow Finger protocol requests (defined in RFC 742) from the network server.
service hide-telnet-addresses	Hide destination addresses in Telnet command.
service linenumber	Enable a line number banner for each EXEC.
service nagle	Enable the Nagle congestion control algorithm.
service old-slip-prompts	Allow old scripts to operate with SLIP/PPP.
service pad	Enable Packet Assembler Dissembler commands.
service password-encryption	Enable encrypt passwords.
service prompt	Enable a mode-specific prompt.
service tcp-keepalives {**in** | **out**}	Configure keepalive packets on idle network connections.
service tcp-small-servers	Enable small TCP servers (for example, ECHO).
service telnet-zero-idle	Set the TCP window to zero (0) when the Telnet connection is idle.
service timestamps	Display timestamp debug and log messages.
service udp-small-servers	Enable small UDP servers (for example, ECHO).

Configuring SNMP Access Policy

To create or update an access policy, use these global configuration commands:

Command	Task
snmp-server access-policy *destination-party source-party context privileges*	Configure global access policy.
snmp-server chassis-id *text*	Provide a message line identifying the SNMP server serial number.
snmp-server community *string* [**RO** | **RW**] [*number*]	Configure the SNMP community access string.
snmp-server contact *text*	Configure the system contact (syscontact) string.
snmp-server context *context-name context-oid view-name*	Configure a context record.
snmp-server host *host community-string* [**envmon**] [**frame-relay**] [**sdlc**] [**snmp**] [**tty**] [**x25**]	Configure the recipient of an SNMP trap operation.
snmp-server location *text*	Configure a system location string.
snmp-server packetsize *byte-count*	Configure the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply.
snmp-server party *party-name party-oid* [*protocol-address*] [**packetsize** *size*] [**local** | **remote**] [**authentication** {**md5** *key* [**clock** *clock*] [**lifetime** *lifetime*] | **snmpv1** *string*}]	Configure a party record.
snmp-server queue-length *length*	Configure the message queue length for each trap host.
snmp-server system-shutdown	Configure SNMP message reload.
snmp-server trap-authentication [**snmpv1** | **snmpv2**]	Configure trap message authentication.
snmp-server trap-timeout *seconds*	Configure the frequency with which to resend trap messages on the retransmission queue.
snmp-server userid *user-id* [**view** *view-name*] [**RO** | **RW**] [**password** *password*]	Configure SNMP v.2 security context using the simplified security conventions method.
snmp-server view *view-name oid-tree* {**included** | **excluded**}	Configure view entry.

To display the SNMP status, use the EXEC command:

Command	Task
show snmp	Check the status of communications between the SNMP agent and SNMP manager.

Establishing Username Commands

To establish a username-based authentication system at login, use the global configuration commands:

Command	Task
username *name* [**no password** | **password** *encryption-type password*]	Configure username-based authentication system at login.
username *name* **password** *secret*	Configure username-based CHAP authentication system at login.
username *name* [**autocommand** *command*]	Configure username-based authentication system at login with an additional command to be added.
username *name* [**noescape**] [**nohangup**]	Configure username-based authentication system at login without escape but with another login prompt.

Configuring the Privilege Level

This section describes how to configure and display the privilege level access to the DSLAM. You can configure access privileges at the global level for the entire DSLAM, or at the line level for a specific line.

Configuring the Global Privilege Level

To set the privilege level for a command, use the global configuration command:

Command	Task
privilege *mode* **level** *level command*	Set the privilege level.

To display your current level of privilege, use the privileged EXEC command:

Command	Task
show privilege	Display the privilege level.

Configuring Privilege Level for a Line

To set the default privilege level for a line, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1	**line** [**aux** | **console** | **vty**] *line-number*	Select the line to configure.
2	**privilege level** *level*	Configure the default privilege level.

To display your current level of privilege, use the privileged EXEC command:

Command	Task
show privilege	Display the privilege level.

Configuring the Network Time Protocol

This section describes how to configure the Network Time Protocol (NTP) on the DSLAM.

To control access to the system NTP services, use the global NTP configuration commands in this section. To remove access control to the system's NTP services, use the **no ntp** command. See the example configuration at the end of this section and the output examples to confirm the NTP configuration.

To view a list of the NTP commands enter a **?** in EXEC configuration mode. This example shows the list of commands available for NTP configuration:

```
DSLAM(config)# ntp ?
  access-group        Control NTP access
  authenticate        Authenticate time sources
  authentication-key  Authentication key for trusted time sources
  broadcastdelay      Estimated round-trip delay
  clock-period        Length of hardware clock tick
  master              Act as NTP master clock
  max-associations    Set maximum number of associations
  peer                Configure NTP peer
  server              Configure NTP server
  source              Configure interface for source address
  trusted-key         Key numbers for trusted time sources
  update-calendar     Periodically update calendar with NTP time
```

To control access to the system NTP services, use the global configuration command:

Command	Task
ntp access-group {**query-only** | **serve-only** | **serve** | **peer**} *access-list-number*	Configure a NTP access group.

To enable NTP authentication, perform these steps in global configuration mode:

Step	Command	Task
1	**ntp authenticate**	Enable NTP authentication.
2	**ntp authentication-key** *number* **md5** *value*	Define an authentication key.

To specify that a specific interface should send NTP broadcast packets, perform these steps, beginning in Global Configuration mode:

Step	Command	Task
1	**interface** *type slot/port*	Select the physical interface to be configured.
2	**ntp broadcastdelay** *microseconds*	Configure the system to receive NTP broadcast packets.

As the NTP compensates for any error in the system clock, it keeps track of the correction factor needed to correct this error. The system automatically saves this correction factor into the system configuration using the **ntp clock-period** global configuration command.

Caution Do not enter the **ntp clock-period** command. It is documented for informational purposes only. The system automatically generates this command as the NTP determines the clock error and compensates.

To prevent an interface from receiving NTP packets, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**interface** *type slot/port*	Select the physical interface to be configured.
2	**ntp disable**	Disable the NTP receive interface.

To configure the DSLAM as a NTP master clock to which peers synchronize themselves when an external NTP source is not available, use the global configuration command:

Command	Task
ntp master [*stratum*]	Configure the DSLAM as a NTP master clock.

To configure the DSLAM as a NTP peer that receives its clock synchronization from an external NTP source, use the global configuration command:

Command	Task
ntp peer *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]	Configure the DSLAM system clock to synchronize a peer or to be synchronized by a peer.

To allow the DSLAM system clock to be synchronized by a time server, use the global configuration command

:

Command	Task
ntp server *ip-address* [**version** *number*] [**key** *keyid*] [**source** *interface*] [**prefer**]	Configure the DSLAM system clock to allow it to be synchronized by a time server.

To use a particular source address in NTP packets, use the global configuration command:

Command	Task
ntp source *interface*	Configure a particular source address in NTP packets.

To authenticate the identity of a system to which the NTP will synchronize, use the global configuration command:

Command	Task
ntp trusted-key *key-number*	Configure a NTP synchronize number.

To periodically update the DSLAM calendar from the NTP, use the global configuration command:

Command	Task
ntp update-calendar	Update a NTP calendar.

Example

This example configures the DSLAM to synchronize its clock and calendar to a NTP server, using Ethernet port 0/0:

```
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# ntp server 198.92.30.32
DSLAM(config)# ntp source Ethernet 0/0
DSLAM(config)# ntp authenticate
DSLAM(config)# ntp max-associations 2000
DSLAM(config)# ntp trusted-key 22507
DSLAM(config)# ntp update-calendar
```

To show the status of NTP associations, use the privileged EXEC commands:

Command	Task
show ntp associations [**detail**]	Display NTP associations.
show ntp status	Display the NTP status.

Examples

This example displays the DSLAM detail NTP configuration:

```
DSLAM# show ntp associations detail
```

```
198.92.30.32 configured, our_master, sane, valid, stratum 3
ref ID 171.69.2.81, time B6C04E67.6E779000 (18:18:15.431 UTC Thu Feb 27 1997)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 109.51 msec, root disp 377.38, reach 377, sync dist 435.638
delay -3.88 msec, offset 7.7674 msec, dispersion 1.57
precision 2**17, version 3
org time B6C04F19.437D8000 (18:21:13.263 UTC Thu Feb 27 1997)
rcv time B6C04F19.41018C62 (18:21:13.253 UTC Thu Feb 27 1997)
xmt time B6C04F19.41E3EB4B (18:21:13.257 UTC Thu Feb 27 1997)
filtdelay =    -3.88   -3.39   -3.49   -3.39   -3.36   -3.46   -3.37   -3.16
filtoffset =    7.77    6.62    6.60    5.38    4.13    4.43    6.28   12.37
filterror =     0.02    0.99    1.48    2.46    3.43    4.41    5.39    6.36
```

This example displays the DSLAM NTP status:

```
DSLAM# show ntp status
Clock is synchronized, stratum 4, reference is 198.92.30.32
nominal freq is 250.0000 Hz, actual freq is 249.9999 Hz, precision is 2**24
reference time is B6C04F19.41018C62 (18:21:13.253 UTC Thu Feb 27 1997)
clock offset is 7.7674 msec, root delay is 113.39 msec
root dispersion is 386.72 msec, peer dispersion is 1.57 msec
```

Configuring the Clock and Calendar

If no other source of time is available, you can manually configure the current time and date after the system is restarted. The time setting remains accurate until the next system restart. Cisco recommends that you use manual configuration only as a last resort.

Note If you have an outside source to which the DSLAM can synchronize, you do not need to manually set the system clock.

Configuring the Clock

To configure, read, and set the DSLAM as a time source for a network based on its calendar, perform these steps in global configuration mode:

Step	Command	Task
1	**clock calendar-valid**	Set the DSLAM as the default clock.
2	**clock summer-time** *zone* **recurring** [*week day month hh:mm week day month hh:mm* [*offset*]]	Configure the system to automatically switch to summer time (daylight savings time), use one of the formats of the **clock summer-time** configuration command. Use the **no** form of this command to configure the DSLAM to not automatically switch to summer time.
3	**clock timezone** *zone*	Configure the system time zone.

To manually read and set the calendar for the DSLAM system clock, perform these steps in privileged EXEC mode:

Step	Command	Task
1	**clock read-calendar**	Manually read the calendar into the DSLAM.
2	**clock set** *hh:mm:ss day month year*	Manually set the system clock.
3	**clock update-calendar**	Set the calendar.

To display the system clock information, use the EXEC command:

Command	Task
show clock [**detail**]	Display the system clock.

Configuring the Calendar

To set the system calendar, use the privileged EXEC command:

Command	Task
calendar set *hh:mm:ss day month year*	Configure the calendar.

To display the system calendar information, use the EXEC command:

Command	Task
show calendar	Display the calendar setting.

Configuring the Terminal Access Control Access System

You can configure the DSLAM to use one of three special TCP/IP protocols related to Terminal Access Controller Access Control System (TACACS): regular TACACS, extended TACACS, or AAA/TACACS+. TACACS services are provided by and maintained in a database on a TACACS server running on a workstation. You must have access to and configure a TACACS server before configuring the TACACS features described in this publication on your Cisco device. Cisco basic TACACS support is modeled after the original Defense Data Network (DDN) application.

A comparative description of the supported versions follows. Table 4-1 compares the versions by commands.

- TACACS—Provides password checking, authentication, and notification of user actions for security and accounting purposes.
- Extended TACACS—Provides information about protocol translator and DSLAM use. This information is used in UNIX auditing trails and accounting files.
- AAA/TACACS+—Provides more detailed accounting information as well as more administrative control of authentication and authorization processes.

You can establish TACACS-style password protection on both user and privileged levels of the system EXEC.

Table 4-1 TACACS Command Comparison

Command	TACACS	Extended TACACS	TACACS+
aaa accounting			X
aaa authentication arap			X
aaa authentication enable default			X
aaa authentication login			X
aaa authentication local override			X
aaa authentication ppp			X
aaa authorization			X
aaa new-model			X
arap authentication			X
arap use-tacacs	X	X	
enable last-resort	X	X	
enable use-tacacs	X	X	
login authentication			X
login tacacs	X	X	
ppp authentication	X	X	X
ppp use-tacacs	X	X	X
tacacs-server attempts	X	X	X
tacacs-server authenticate	X	X	
tacacs-server extended		X	
tacacs-server host	X	X	X
tacacs-server key			X
tacacs-server last-resort	X	X	
tacacs-server notify	X	X	
tacacs-server optional-passwords	X	X	
tacacs-server retransmit	X	X	X
tacacs-server timeout	X	X	X

Enabling TACACS and Extended TACACS

This section describes the features available with TACACS and Extended TACACS. The Extended TACACS software is available using FTP (see the README file in the *ftp.cisco.com* directory).

Note You cannot use several original TACACS and extended TACACS commands after you initialize AAA/TACACS+. To identify which commands you can use with the three versions, refer to Table 4-1.

These sections describe TACACS configuration:

- Configuring AAA Access Control with TACACS+
- Configuring AAA Accounting
- Configuring a TACACS Server
- Configuring PPP Authentication

Configuring AAA Access Control with TACACS+

To enable the AAA access control model that includes TACACS+, use the global configuration command:

Command	Task
aaa new-model	Enable the AAA access control model.

Configuring AAA Accounting

To enable the AAA accounting of requested services for billing or security purposes when using TACACS+, perform these steps in global configuration mode:

Step	Command	Task
1	**aaa accounting system**	Perform accounting for all system-level events not associated with users, such as reloads.
2	**aaa accounting network**	Run accounting for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARAP.
3	**aaa accounting connection**	Run accounting for outbound Telnet and rlogin.
4	**aaa accounting exec**	Run accounting for Execs (user shells). This keyword might return user profile information such as **autocommand** information.
5	**aaa accounting command**	Run accounting for all commands at the specified privilege level.
6	**start-stop tacacs+**	Send a start record accounting notice at the beginning of a process and a stop record at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether or not the accounting server receives the start accounting record.

Step	Command (continued)	Task
7	**wait-start tacacs+**	As in **start-stop**, sends both a start and a stop accounting record to the accounting server. However, if you use the **wait-start** keyword, the requested user service does not begin until the start accounting record is acknowledged. A stop accounting record is also sent.
8	**stop-only tacacs+**	Send a stop record accounting notice at the end of the requested user process.

Configuring a TACACS Server

To configure a TACACS server, perform these steps in global configuration mode:

Step	Command	Task
1	**tacacs-server attempts** *count*	Configure the number of login attempts allowed.
2	**tacacs-server authenticate** {**connection** [**always**] | **enable** | **slip** [**always**] [**access-lists**]}	Configure if the user may perform an action.
3	**tacacs-server extended**	Configure extended TACACS mode.
4	**tacacs-server host** *name*	Configure a TACACS host.
5	**tacacs-server last-resort** {**password** | **succeed**}	Configure a network server to request a privileged password as verification.
6	**tacacs-server notify** {**connection** [**always**] | **enable** | **logout** [**always**] | **slip** [**always**]}	Configure transmission to the TACACS server.
7	**tacacs-server optional-passwords**	Configure the initial TACACS request to a TACACS server to be made *without* password verification.
8	**tacacs-server retransmit** *retries*	Configure the number of times the system software will search the list of TACACS server hosts.
9	**tacacs-server timeout** *seconds*	Configure the interval that the server waits for a server host to reply.

Configuring PPP Authentication

To enable Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) and to enable an AAA authentication method on an interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**interface** *type slot/port*	Select the physical interface to be configured.

Step	Command	Task
2	**ppp authentication** {**chap** | **pap**} [**if-needed**] [**list-name**]	Configure PPP authentication.
3	**ppp use-tacacs** [**single-line**]	Enable the PPP authentication for TACACS.

To enable TACACS to determine whether a user can access the privileged command level, use the global configuration command:

Command	Task
enable use-tacacs	Enable TACACS.

Testing the System Management Functions

This section describes the commands you use to monitor and display the system management functions.

Showing Active Processes

To display information about the active processes, use the privileged EXEC commands:

Command	Task
show processes [**cpu**]	Display active processes.
show processes memory	Display memory utilization.

Showing Protocols

To display the configured protocols, use the privileged EXEC command:

Command	Task
show protocols	Display the global and interface-specific status of any configured Level 3 protocol; for example, IP, DECnet, Internet Packet Exchange (IPX), and AppleTalk.

Showing Stacks

To monitor the stack utilization of processes and interrupt routines, use the privileged EXEC command:

Command	Task
show stacks	Display system stack trace information.

The **show stacks** display includes the reason for the last system reboot. If the system was reloaded because of a system failure, a saved system stack trace is displayed. This information is of use only to Cisco engineers analyzing system failure in the field. It is included here in case you need to read the displayed statistics to an engineer over the telephone.

Showing Routes

To discover the IP routes that the switch packets take when the packets travel to their destination, use the EXEC command:

Command	Task
traceroute [*protocol*] [*destination*]	Display switch packets through the network.

Showing Temperature and Voltage Information

To display temperature and voltage information on the DSLAM console, use the EXEC commands:

Command	Task
show environment	Display temperature and voltage information.
show environment all	Display all temperature and voltage information.
show environment last	Display the last logs of the last measured value from each of the six test points to internal nonvolatile memory.
show environment table	Display environmental measurements and a table that lists the ranges of environment measurement.

Checking Basic ATM and IP Network Connectivity

To diagnose basic ATM and IP network connectivity, use the privileged EXEC command:

Command	Task
ping atm interface atm *slot/port*[*.vpt*] *vpi vci*	Use **ping** to check the ATM network connection.

CHAPTER 5

Configuring Virtual Connections

This chapter describes how to configure virtual connections (VCs) in a typical ATM network after autoconfiguration has established the default network connections. The network configuration modifications described in this chapter are used to optimize your ATM network operation.

Characteristics and Types and of Virtual Connections

The characteristics of the VC, established when the VC is created, include:

- Quality of service (QoS)
- ATM adaption layer 5 (AAL5)
- Peak and average transmission rates
- Cell sequencing integrity

These switching features can be turned off with interface configuration commands; autonomous switching must be explicitly enabled per interface. SVC connection setup is possible both on trunk/subtended interfaces as well as all subscriber ports.

The total number of SVCs supported is approximately 24k. The total number of PVCs supported is approximately 5k (constrained by Flash size). The call rate is two hundred calls per second.

Table 5-1 lists the types of supported virtual connections.

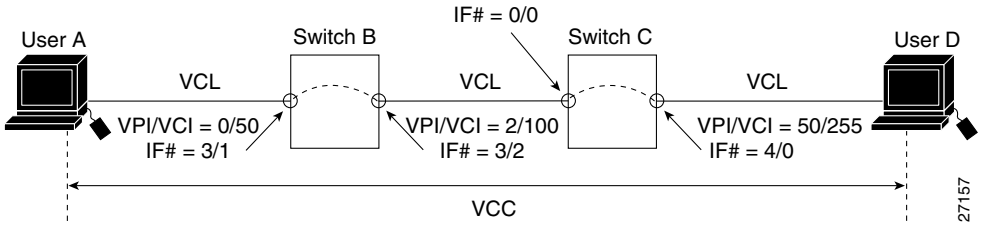
Table 5-1 Supported DSLAM Virtual Connection Types

Connection	Point-to-Point	Transit	Terminate
Permanent virtual channel link (PVCL)	3	—	—
Permanent virtual path link (PVPL)	3	—	—
Permanent virtual channel (PVC)	3	3	3
Permanent virtual path (PVP)	3	3	—
Soft permanent virtual channel (Soft PVC)	3	3	—
Soft permanent virtual path (Soft PVP)	3	3	—
Switched virtual circuit (SVC)	3	3	3
Switched virtual path (SVP)	3	3	—

Configuring Permanent Virtual Channel Connections

This section describes how to configure DSLAM VCCs. A VCC is established as a bidirectional facility to transfer ATM traffic between two ATM layer users. Figure 5-1 shows an example VCC between ATM user A and user D.

Figure 5-1 Virtual Channel Connection Example



Note The value of the VPIs and VCIs can change as the traffic is relayed through the ATM network.

To configure a point-to-point VCC, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.sub-inter#*]	Select the interface to be configured.
2.	**atm pvc** *vpi* [*vci* | **any-vci**[1]] [**upc** *upc*] [**pd** *pd*[2]] [**rx-cttr** *index*] [**tx-cttr** *index*] **interface atm***slot/port*[*.vpt#*] *vpi* [*vci* | **any-vci**] [**upc** *upc*]	Configure the PVC.

1. The **any-vci** parameter is only available for ATM interface0/0.
2. The parameter *pd* is not applicable to a virtual path.

Note You must configure the row index for **rx-cttr** and **tx-cttr** before you use this optional parameter. See Chapter 9, "Configuring Resource Management."

Note When you configure PVC connections, configure the lowest VPI and VCI numbers first.

Examples

This example shows how to configure the internal cross-connect PVC on DSLAM B between interface 0/1, VPI = 0, VCI = 50 and interface 0/2, VPI = 2, VCI = 100 (see Figure 5-1):

```
DSLAM-B(config)# interface atm 0/1
DSLAM-B(config-if)# atm pvc 0 50 interface atm 0/2 2 100
```

This example shows how to configure the internal cross-connect PVC on DSLAM C between interface 1/0, VPI = 2, VCI = 100 and interface0/1, VPI 50, VCI = 255:

```
DSLAM-C(config)# interface atm 1/0
DSLAM-C(config-if)# atm pvc 2 100 interface atm 0/1 50 255
```

Each subsequent VC cross-connection and link must be configured until the VC is terminated to create the entire VCC.

This example shows how to configure the CPU leg of any terminating PVC:

```
DSLAM(config)# interface atm0/0
DSLAM(config-if)# atm pvc 0 ?
  <32-16383>  vci
  any-vci     Choose any available vci

DSLAM(config-if)# atm pvc 0 any-vci
DSLAM(config-if)#
```

When configuring the CPU leg of a PVC that is not a tunnel, the VPI should be configured as 0. The preferred method of VCI configuration is to select the **any-vci** parameter, unless a specific VCI is needed as a parameter in another command, such as **map-list**.

Note If configuring a specific VCI value for the CPU leg, select a VCI value higher than 300. This step prevents a conflict with an automatically assigned VCI for well-known channels if the DSLAM reboots.

To show the VC configuration, use these EXEC commands:

Step	Command	Task
1.	**show atm interface** [**atm** *slot/port*]	Show the ATM interface configuration.
2.	**show atm vc** [**interface atm** *slot/port vpi vci*]	Show the PVC interface configuration.

Examples

This example displays the DSLAM B PVC configuration on ATM interface 0/2:

```
DSLAM-B# show atm interface 0/2

Interface:      ATM0/2          Port-type:    suni-dual
IF Status:      UP              Admin Status: up
Auto-config:    enabled         AutoCfgState: completed
IF-Side:        Network         IF-type:      NNI
Uni-type:       not applicable  Uni-version:  not applicable
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0001.0000.0001.4000.0c81.8010.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      5        0       0       0        0       0           5                5
Logical ports(VP-tunnels):     0
Input cells:    527452          Output cells: 527485
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 344372, Output AAL5 pkts: 344384, AAL5 crc errors: 0
```

This example displays the DSLAM B PVC configuration on ATM interface 0/2:

```
DSLAM-B# show atm vc interface atm 0/2
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM0/2     0     5      PVC     ATM0/0     0     57    QSAAL  UP
```

```
ATM0/2     0     16     PVC     ATM0/0     0     37    ILMI   UP
ATM0/2     0     18     PVC     ATM0/0     0     73    PNNI   UP
ATM0/2     0     50     PVC     ATM0/0     2     100          UP
ATM0/2     1     50     PVC     ATM0/0     0     80    SNAP   UP
```

This example displays the DSLAM B configuration on interface 0/1, VPI = 0, VCI = 50:

```
DSLAM-B# show atm vc interface atm 0/1 0 50

Interface: ATM0/1, Type: suni-dual
VPI = 0  VCI = 50
Status: UP
Time-since-last-status-change: 00:31:30
Connection-type: PVC
Cast-type: point-to-point
Packet-discard-option: disabled
Usage-Parameter-Control (UPC): pass
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/2, Type: suni-dual
Cross-connect-VPI = 2
Cross-connect-VCI = 100
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Rx cells: 0, Tx cells: 0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

This example displays the DSLAM B PVC configuration on ATM interface 0/2, VPI = 0, VCI = 50:

```
DSLAM-B# show atm vc interface atm 0/2 0 50

Interface: ATM0/2, Type: suni-dual
VPI = 0  VCI = 50
Status: UP
Time-since-last-status-change: 4d02h
Connection-type: PVC
Cast-type: point-to-point
Packet-discard-option: disabled
Usage-Parameter-Control (UPC): pass
Wrr weight: 32
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/1, Type: suni-dual
Cross-connect-VPI = 2
Cross-connect-VCI = 100
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
```

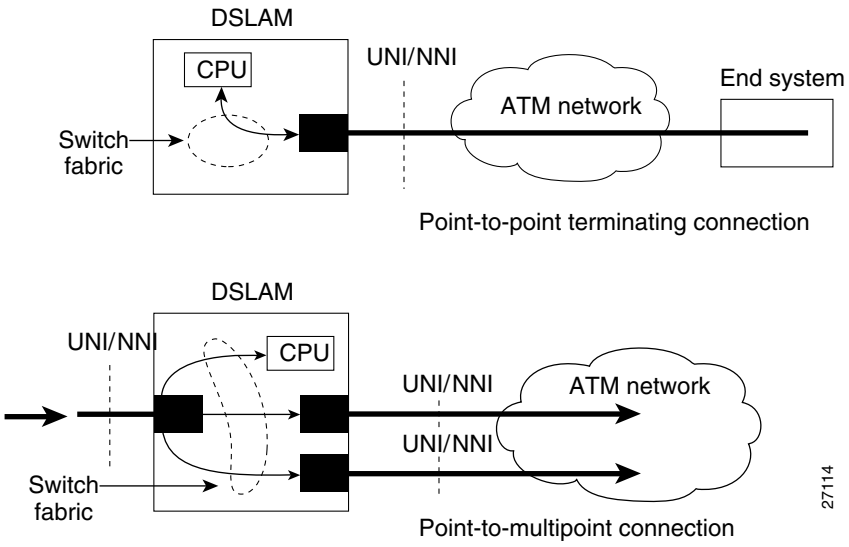
```
Threshold Group: 5, Cells queued: 0
Rx cells: 0, Tx cells: 0
Tx Clp0:0,  Tx Clp1: 0
Rx Clp0:0,  Rx Clp1: 0
Rx Upc Violations:0, Rx cell drops:0
Rx Clp0 q full drops:0, Rx Clp1 qthresh drops:0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

Configuring Terminating PVC Connections

This section describes how to configure point-to-point terminating PVC connections. Terminating connections provide the connection to the DSLAM CPU for control channels for Interim Local Management Interface (ILMI), signaling, and Private Network-to-Network Interface (PNNI) plus network management.

Figure 5-2 shows an example of transit and terminating connections.

Figure 5-2 Virtual Connection Types Example



Point-to-point is a type of terminating connection. Terminating connections are configured using the same commands as transit connections (discussed in the previous sections). However, all switch terminating connections use interface 0/0 to connect to the switch CPU.

To configure point-to-point terminating PVC connections, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot-A/port-A*[*.vpt#*]	Select the interface to be configured.
2.	**atm pvc** *vpi-A* [*vci-A* | **any-vci**[1]] [**cast-type** *p2mp-leaf p2mp-root p2p*] [**upc** *upc-A*] [**pd** *pd*] [**rx-cttr** *index*] [**tx-cttr** *index*] **interface atm** *slot-B/subslot-B/port-B*[*.vpt#*] *vpi-B* [*vci-A* | **any-vci**] [**upc** *upc-B*] [**cast-type** *p2mp-leaf* | *p2mp-root* | *p2p*]	Configure the PVC between ATM switch connections.

1. The **any-vci** feature is only available for interface ATM 0/0.

Note You must configure the row index for **rx-cttr** and **tx-cttr** before you use this optional parameter. See Chapter 9, "Configuring Resource Management."

Examples

This example shows how to configure the CPU leg of any terminating PVC:

```
DSLAM(config)# interface atm0/0
DSLAM(config-if)# atm pvc 0 ?
  <32-16383>  vci
  any-vci     Choose any available vci

DSLAM(config-if)# atm pvc 0 any-vci
```

When configuring the CPU leg of a PVC that is not a tunnel, the VPI should be configured as 0. The preferred method of VCI configuration is to select the **any-vci** parameter, unless a specific VCI is needed as a parameter in another command, such as **map-list**.

Note If you configure a specific VCI value for the CPU leg, select a VCI value greater than 300 to prevent a conflict with an automatically assigned VCI for well-known channels if the DSLAM reboots.

This example shows how to configure the internal cross-connect PVC between interface 0/2, VPI = 1, VCI = 50 and the terminating connection at the CPU interface 0/0, VPI = 0, and VCI unspecified:

```
DSLAM-B(config)# interface atm 0/2
DSLAM-B(config-if)# atm pvc 1 50 interface atm0/0 0 any-vci encap aal5snap
```

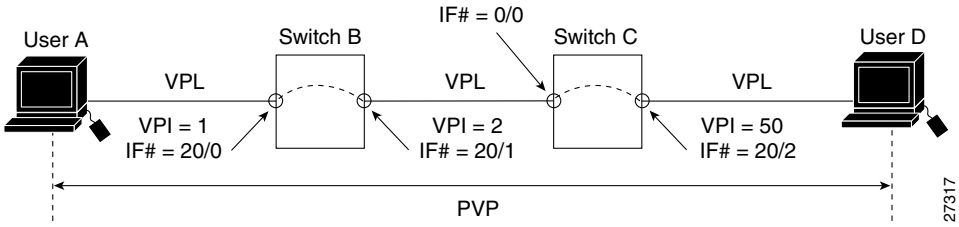
To show the terminating PVC configuration, use the EXEC commands:

Step	Command	Task
1.	**show atm vc** *slot/port*	Show the ATM interface configuration.
2.	**show atm vc interface atm** *slot/port vpi vci*	Show the PVC interface configuration.

Configuring Permanent Virtual Path Connections

This section describes how to configure a PVP connection. Figure 5-3 shows an example of a DSLAM with PVPs configured through the switch. Switch B and Switch C can be either ATM switches or DSLAMs.

Figure 5-3 Virtual Path Connection Example



To configure a PVP connection, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the physical interface to be configured.
2.	**atm pvp** *vpi* [**cast-type** *type*] [**upc** *upc*] [**rx-cttr** *index*] [**shaped**] [**tx-cttr** *index*] **interface** *slot/port*	Configure the interface PVP.

Note You must configure the row index for **rx-cttr** and **tx-cttr** before you use this optional parameter. See Chapter 9, "Configuring Resource Management."

Note No traffic shaping or policing is available in the downstream direction.

Note When you configure PVC connections, configure the lowest VPI and VCI numbers first.

Examples

This example shows how to configure the internal cross-connect PVP within DSLAM B between interfaces 0/2, VPI = 1 and interface 0/1, VPI = 2:

```
DSLAM-B# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM-B(config)# interface atm 0/2
DSLAM-B(config-if)# atm pvp 1 interface atm 0/1 2
```

This example shows how to configure the internal cross-connect PVP within DSLAM C between interfaces 0/1, VPI = 2 and interface 0/2, VPI = 50:

```
DSLAM-C(config)# interface atm 0/1
DSLAM-C(config-if)# atm pvp 2 interface atm 0/2 50
```

Each subsequent PVP cross connection and link must be configured until the VP is terminated to create the entire PVP.

To show the ATM interface configuration, use this EXEC command:

Command	Task
show atm vp [**interface atm** *slot/port vpi*]	Show the ATM VP configuration.

Example

This example displays the PVP configuration of DSLAM B:

```
DSLAM-B# show atm vp
Interface     VPI    Type  X-Interface     X-VPI     Status
ATM0/1        1      PVP   ATM0/2          2         UP
```

This example displays the PVP configuration of DSLAM B:

```
DSLAM-B# show atm vp interface atm 0/1 1

Interface: ATM0/1, Type: suni-dual
VPI = 1
Status: TUNNEL
Time-since-last-status-change: 4d22h
Connection-type: PVP
Cast-type: point-to-point
Usage-Parameter-Control (UPC): pass
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/2, Type: suni-dual
Cross-connect-VPI = 2
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Rx cells: 0, Tx cells: 0
Rx Upc Violations:0, Rx cell drops:0
Rx Clp0 q full drops:0, Rx Clp1 qthresh drops:0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

This example displays the PVP configuration of DSLAM B:

```
DSLAM-B# show atm vp interface atm 0/1 1

Interface: ATM3/1/0, Type: suni-dual
VPI = 1
Status: TUNNEL
```

```
Time-since-last-status-change: 4d22h
Connection-type: PVP
Cast-type: point-to-point
Usage-Parameter-Control (UPC): pass
Wrr weight: 32
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/2, Type: suni-dual
Cross-connect-VPI = 2
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Threshold Group: 5, Cells queued: 0
Rx cells: 0, Tx cells: 0
Tx Clp0:0,  Tx Clp1: 0
Rx Clp0:0,  Rx Clp1: 0
Rx Upc Violations:0, Rx cell drops:0
Rx Clp0 q full drops:0, Rx Clp1 qthresh drops:0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

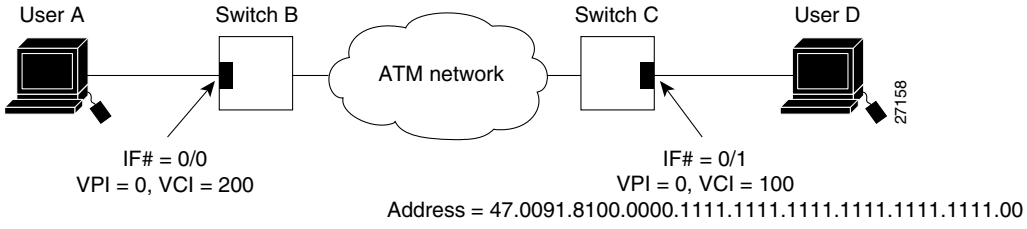
Configuring Soft PVC Connections

This section describes how to configure soft PVC connections, which provide these features:

- Connection to another host, switch, or DSLAM that does not support signaling
- Configuration of PVCs without the manual configuration steps described in the "Configuring Permanent Virtual Channel Connections" section on page 5-60.
- Configuration of PVCs with the reroute or retry capabilities if a failure occurs in the network

Figure 5-4 illustrates the soft PVC connections used in these examples. Switch B and Switch C can be ATM switches or DSLAMs.

Figure 5-4 Soft Permanent Virtual Channel Connection Example



Guidelines for Creating Soft PVCs

Perform these steps when you configure *soft PVCs*:

Step 1 Determine which two ports you want to define as participants in the soft PVC.

Step 2 Decide which of these two ports you want to designate as the destination (or passive) side of the soft PVC.

This decision is arbitrary—it makes no difference which port you define as the destination end of the circuit.

Step 3 Configure the destination (passive) side of the soft PVC.

You must configure the destination end of the soft PVC first, to define an ATM address for that port.

You must retrieve this address (see Step 4), and the VPI/VCI values for the circuit (see Step 5), and use these elements as part of the command string when you configure the source (active) end of the soft PVC (see Step 6).

Step 4 Retrieve the ATM address of the destination end of the soft PVC using the **show atm address** command. This command typically produces output in the form:

```
DSLAM# show atm address

Switch Address(es):
  47.00918100000000400B0A2A81.00400B0A2A81.00 active

Soft VC Address(es):
  47.0091.8100.1111.1111.1111.1111.1111.1111.1111.00 ATM4/0

ILMI Switch Prefix(es):
  47.0091.8100.0000.0040.0b0a.2a81

ILMI Configured Interface Prefix(es):

LECS Address(es):
```

Step 5 Retrieve the VPI/VCI values for the circuit using the **show atm vc** command. This command typically produces output in the form:

```
DSLAM# show atm vc interface atm 0/0
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM0/0     0     5      PVC     ATM0/0     0     52    QSAAL  DOWN
ATM0/0     0     16     PVC     ATM0/0     0     32    ILMI   DOWN
ATM0/0     0     200    SoftVC  ATM0/0     0     100          UP
DSLAM#
```

Step 6 Configure the source (active) end of the soft PVC. At the same time, complete the soft PVC setup using the information derived from Step 4 and Step 5.

You must configure the source end of the soft PVC last because this not only defines the configuration information for the source port, but also requires you to enter the ATM address and VPI/VCI values for the destination port.

If you have not already defined the destination port for the soft PVC (as required in Step 3), this ATM address is not defined for the destination port, and the VPI/VCI values are not available, as required in Step 6 for use in completing the soft PVC.

Configuring Soft Permanent Virtual Channels

To configure a soft PVC connection, perform these steps, beginning in privileged EXEC mode:

Step	Command	Task
1.	**show atm addresses**	Determine the destination ATM address.
2.	**configure terminal**	At the privileged EXEC prompt, enter configuration mode from the terminal.
3.	**interface atm** *slot/port* [*.vpt#*]	Select the interface to be configured.
4.	**atm soft-vc** *src-vpi src-vci* **dest-address** *dest_address dest-vpi dest-vci* [**pd** *pd*] [**rx-cttr** *index*] [**slow-retry-interval** *seconds*] [**tx-cttr** *index*] [**upc** *drop pass tag*]	Configure the soft PVC connection.

Note You must configure the row index for **rx-cttr** and **tx-cttr** before you use this optional parameter. See Chapter 9, "Configuring Resource Management."

Examples

This example shows how to allow User A to determine the destination ATM address of the interface connected to User D:

```
Switch-C# show atm addresses

Switch Address(es):
  47.00918100000000603E5ADB01.00603E5ADB01.00 active

Soft VC Address(es):
  47.0091.8100.1111.1111.1111.1111.1111.1111.1111.00 ATM0/0

ILMI Switch Prefix(es):
  47.0091.8100.0000.0060.3e5a.db01

ILMI Configured Interface Prefix(es):

LECS Address(es):
```

This example shows how to configure a soft PVC on Switch B between interface 0/0, source VPI = 0, source VCI = 200; and switch C, destination ATM address = 47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00, VPI = 0, VCI = 100 (see Figure 5-4):

```
Switch-B(config)# interface atm 0/0
Switch-B(config-if)# atm soft-vc 0 200 dest-address
47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00 0 100
```

To display the soft VC configuration at either end of a switch or DSLAM, use the EXEC commands:

Step	Command	Task
1.	**show atm vc** *slot/port*	Show the ATM interface configuration.
2.	**show atm vc** [**interface atm** *slot/port vpi vci*]	Show the soft VC interface configuration.

Examples

This example displays the soft VC configuration of Switch B, on interface 0/0 out to the ATM network:

```
Switch-B# show atm vc interface atm 0/0
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM0/0     0     5      PVC     ATM0/0     0     52    QSAAL  DOWN
ATM0/0     0     16     PVC     ATM0/0     0     32    ILMI   DOWN
ATM0/0     0     200    SoftVC  ATM0/0     0     100          UP
```

This example displays the soft VC configuration of Switch C, on interface 4/0 out to the ATM network:

```
Switch-C# show atm vc interface atm 4/0
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM4/0       0     5      PVC     ATM0/0     0     52    QSAAL  DOWN
ATM4/0       0     16     PVC     ATM0/0     0     32    ILMI   DOWN
ATM4/0       0     100    SoftVC  ATM0/0     0     200          UP
```

This example displays the soft VC configuration of Switch B, on interface 0/0, VPI = 0, VCI = 200 out to the ATM network:

```
Switch-B# show atm vc interface atm 0/0 0 200
Interface: ATM0/0, Type: suni-dual
VPI = 0  VCI = 200
Status: NOT CONNECTED
Time-since-last-status-change: 00:00:45
Connection-type: SoftVC
Cast-type: point-to-point
Soft vc location: Source
Remote ATM address: 47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00
Remote VPI: 0
Remote VCI: 100
Soft vc call state: Inactive
Number of soft vc re-try attempts: 4
Slow-retry-interval: 60 seconds
Next retry in: 29 seconds
Aggregate admin weight: 0
Packet-discard-option: disabled
Usage-Parameter-Control (UPC): pass
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Rx cells: 0, Tx cells: 0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
```

```
Tx       mbs: none
```

This example displays the soft VC configuration of Switch B, on interface 0/0, VPI = 0, VCI = 200 out to the ATM network with FC-PFQ installed:

```
Switch-B# show atm vc interface atm 0/0 0 200
Interface: ATM0/0, Type: suni-dual
VPI = 0  VCI = 200
Status: NOT CONNECTED
Time-since-last-status-change: 00:00:45
Connection-type: SoftVC
Cast-type: point-to-point
Soft vc location: Source
Remote ATM address: 47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00
Remote VPI: 0
Remote VCI: 100
Soft vc call state: Inactive
Number of soft vc re-try attempts: 4
Slow-retry-interval: 60 seconds
Next retry in: 29 seconds
Aggregate admin weight: 0
Packet-discard-option: disabled
Usage-Parameter-Control (UPC): pass
Wrr weight: 32
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Threshold Group: 5, Cells queued: 0
Rx cells: 0, Tx cells: 0
Tx Clp0:0,  Tx Clp1: 0
Rx Clp0:0,  Rx Clp1: 0
Rx Upc Violations:0, Rx cell drops:0
Rx Clp0 q full drops:0, Rx Clp1 qthresh drops:0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

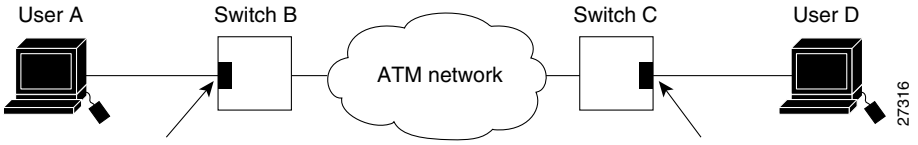
Configuring Soft PVP Connections

This section describes how to configure soft PVP connections. Soft PVC connections provide these features:

- Connection to another host or switch that does not support signaling
- Configuration of PVPs without the manual configuration steps described in this section.
- Configuration of PVPs with the reroute or retry capabilities when a failure occurs within the network

Figure 5-5 is an illustration of the soft PVP connections used in the examples in this section. Switch B and Switch C can be ATM switches or DSLAMs.

Figure 5-5 Soft Permanent Virtual Path Connection Example



To configure a soft PVP connection, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm soft-vp** *src-vpi* **dest-address** *dest-address dest-vpi* [**pd** *pd*] [**rx-cttr** *index*] [**slow-retry-interval** *seconds*] [**tx-cttr** *index*] [**upc** *drop pass tag*]	Configure the soft PVP connection.

Note You must configure the row index for **rx-cttr** and **tx-cttr** before you use this optional parameter. See Chapter 9, "Configuring Resource Management."

Example

This example shows how to configure a soft PVP on Switch B between interface 0/0, source VPI = 75, and Switch C, destination ATM address = 47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00, VPI = 50 (Figure 5-5):

```
Switch-B(config)# interface atm 0/0
Switch-B(config-if)# atm soft-vp dest-address 75
47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00 50
```

To show the ATM virtual path configuration, use this EXEC command:

Command	Task
show atm vp [**interface atm** *slot/port vpi*]	Show the ATM VP configuration.

Examples

This example displays the soft VP configuration at Switch B, on interface 0/0 out to the ATM network:

```
Switch-B# show atm vp
Interface     VPI     Type  X-Interface     X-VPI     Status
ATM0/0        75      SoftVP   ATM4/0         50        UP
```

This example displays the soft VP configuration at Switch C, on interface 4/0 out to the ATM network:

```
Switch-C# show atm vp
Interface     VPI     Type  X-Interface     X-VPI     Status
ATM4/0        50      SoftVP   ATM0/0      75        UP
```

This example displays the soft VP configuration at Switch B, on interface 0/0, VPI = 75 out to the ATM network:

```
Switch-B# show atm vp interface atm 0/0 75

Interface: ATM0/0, Type: suni-dual
VPI = 75
Status: TUNNEL
Time-since-last-status-change: 00:01:10
Connection-type: SoftVP
Cast-type: point-to-point
Soft vp location: Source
Remote ATM address: 47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00
Remote VPI: 50
Soft vp call state: Inactive
Number of soft vp re-try attempts: 4
Slow-retry-interval: 60 seconds
Next retry in: 4 seconds
Aggregate admin weight: 0
Usage-Parameter-Control (UPC): pass
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Rx cells: 0, Tx cells: 0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

This example displays the soft VP configuration at Switch B, on interface 0/0, VPI = 75 out to the ATM network:

```
Switch-B# show atm vp interface atm 0/0 75

Interface: ATM0/0, Type: suni-dual
VPI = 75
Status: TUNNEL
Time-since-last-status-change: 00:01:10
Connection-type: SoftVP
Cast-type: point-to-point
Soft vp location: Source
Remote ATM address: 47.0091.8100.00.0000.1111.1111.1111.1111.1111.1111.00
Remote VPI: 50
Soft vp call state: Inactive
Number of soft vp re-try attempts: 4
Slow-retry-interval: 60 seconds
Next retry in: 4 seconds
Aggregate admin weight: 0
Usage-Parameter-Control (UPC): pass
Wrr weight: 32
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Threshold Group: 5, Cells queued: 0
```

```
Rx cells: 0, Tx cells: 0
Tx Clp0:0,  Tx Clp1: 0
Rx Clp0:0,  Rx Clp1: 0
Rx Upc Violations:0, Rx cell drops:0
Rx Clp0 q full drops:0, Rx Clp1 qthresh drops:0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx tolerance: 1024 (from default for interface)
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx tolerance: none
```

Configuring Non-Default Well-Known PVCs

Normally the default well-known VCs are automatically created with default VCIs. However, for the unusual instances where the DSLAM interfaces with nonstandard equipment, you can configure nondefault well-known VCI values on a per-interface basis.

Table 5-2 lists the default well-known VCs and their default configuration.

Table 5-2 Well-Known Virtual Channels

Channel Type	Virtual Path Identifier	Virtual Channel Identifier
Signaling	0	5
ILMI	0	16
PNNI	0	18
Tag switching	0	32

Caution Do not change the well-known channels to use a VC where the remote end is sending AAL5 messages not intended for the well-known VC. For example, do not swap VC values between two types of well-known VCs.

Overview of Non-Default PVC Configuration

This section provides an overview of the steps you need to perform to configure non-default well-known VCs:

Step 1 Enable manual well-known VC configuration.

Step 2 Delete any existing automatically created well-known VCs.

Step 3 Configure the individual encapsulation type as:

- Signaling (QSAAL)

- ILMI
- PNNI

Step 4 Copy running-configuration file to startup-configuration file.

Configuring Non-Default PVCs

To configure the non-default VCs for signaling, ILMI, and PNNI, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm manual-well-known-vc** {**keep** | **delete**}	Enter manual-well-known-vc mode.
3.	**atm pvc** *vpi vci* [**upc** *upc*] [**pd** *pd*] [**rx-cttr** *index*] [**tx-cttr** *index*] **interface atm0/0 any-vci [encap {ilmi | pnni | qsaal}]**	Configure the nondefault PVC for each signaling and ILMI channel.
4.	**exit**	Exit from interface configuration mode.
5.	**end**	Exit from global configuration mode.
6.	**copy running-config startup-config**	Copy the running configuration file to the startup configuration file.

Note An error condition occurs if either the signaling or ILMI well-known VCs remain unconfigured when an interface is enabled.

Example

This example shows the non-default VC configuration steps:

Step 1 Use the **show atm vc interface atm 0/0** to display the configuration of the existing default well-known VCs for ATM interface 0/0.

Step 2 Change to interface configuration mode for ATM interface 0/0.

Step 3 Enter manual well-known-vc mode and delete the existing default well-known VCs using the **atm manual-well-known-vc delete** command.

Step 4 Confirm deletion by entering **y.**

Step 5 Configure the non-default VC for signaling from 5 (the default) to 35 using the **atm pvc 1 35 interface atm0/0 0 any-vci encap qsaal** command.

Step 6 Configure the ILMI VC, then configure the PNNI VC if needed using the same procedure.

Step 7 Save the new running configuration to the startup configuration using the **copy running-config startup-config** command.

An example of this procedure is:

```
DSLAM# show atm vc interface atm 0/0
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM0/0     0     5      PVC     ATM0/0     0     49    QSAAL  UP
ATM0/0     0     16     PVC     ATM0/0     0     33    ILMI   UP
ATM0/0     0     18     PVC     ATM0/0     0     65    PNNI   UP
DSLAM#
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# atm manual-well-known-vc delete

Okay to delete well-known VCs for this interface? [no]: y
DSLAM(config-if)# exit
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# atm pvc 1 35 interface atm0/0 0 any-vci encap qsaal
DSLAM(config-if)# ^Z
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM# show atm vc interface atm 0/0
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
ATM0/0     1    35     PVC     ATM0/0     0     150   QSAAL  UP
DSLAM# copy running-config startup-config
Building configuration...
[OK]
```

CHAPTER 6

Configuring Operation, Administration, and Maintenance

This chapter describes the Operation, Administration, and Maintenance (OAM) implementation on Cisco DSLAMs, and includes these sections:

OAM Overview

OAM performs fault management and performance management functions at the ATM management (M)-plane layer. The hardware provides both OAM cell filtering and OAM cell insertion support. The filtering and insertion capability is available on each configured circuit.

OAM cell processing is compliant with I.610; however, only fault management and loopback capabilities are available.

Note Current OAM implementation supports only the fault management function, which includes connectivity verification and alarm surveillance.

The DSLAM supports these ATM OAM cell flows:

- F4 flows—OAM information flows between network elements (NEs) used within virtual paths to report an unavailable path or a virtual path (VP) that cannot be guaranteed. Segment flows are processed, as well as end-to-end flows that terminate in the management processor.
- F5 flows—OAM information flows between network elements (NEs) used within virtual connections to report degraded virtual channel (VC) performance such as late arriving cells, lost cells, and cell insertion problems. Segment flows are processed, as well as end-to-end flows that terminate in the management processor.

Both F4 and F5 flows can be configured as either end-to-end or segment-loopback and used with alarm indication signal (AIS) and remote defect indication (RDI) functions.

Note Cells can be sent either on demand or periodically to verify link and connection integrity.

In addition to the standard OAM functions, the DSLAM can also send OAM pings. OAM cells containing the ATM node addresses or IP addresses of intermediate switches allow network administrators to determine the integrity of a chosen connection at any intermediate point along the connection, allowing for network connection debugging and troubleshooting.

OAM software implements ATM Layer F4 and F5 OAM fault management functions. OAM performs standard loopback (end-to-end or segment) and fault detection and notification (AIS and RDI) for each connection. It also maintains a group of timers for the OAM functions. When there is an OAM state change such as loopback failure, OAM software notifies the connection management software. The network operator can enable or disable OAM operation for these switch components:

- The entire switch
- A specific ATM interface
- Each ATM connection

If OAM operation is disabled, outgoing OAM cells are not generated, and all incoming OAM cells are discarded.

To support various OAM operations, the DSLAM hardware provides OAM cell routing functions on a per-connection basis for each direction. These sections describe the OAM tasks:

Configuring OAM Functions

This section describes OAM commands in EXEC, global, and interface configuration mode.

Configure OAM for the Entire Switch

To enable OAM operations for the entire switch, use the global configuration command **atm oam**.

Note These configuration commands are not stored in the nonvolatile RAM (NVRAM).

Command	Task
atm oam [**ais**] [**end-loopback**] [**intercept end-to-end**] [**max-limit** *number*] [**rdi**] [**seg-loopback**]	OAM operations are enabled or disabled with respect to entire switch. The **atm oam rdi** command enables OAM RDI functionality, only on connections that are terminated on the SAR port of the system. If no connection is terminated on the system, this command does not take any effect. If a connection is terminating and it has OAM RDI enabled, then the end-to-end OAM RDI cell is generated and sent out on the connection in response to the received AIS cell.

Note The number of maximum OAM configured connections allowed ranges from 1 to 3200; the default is 3200.

Examples

This example shows how to enable AIS and segment loopback for the entire switch:

```
DSLAM(config)# atm oam ais seg-loopback
% OAM: Switch level seg loopback is enabled

% OAM: Switch level ais is enabled
```

This example shows how to configure the ATM OAM connection maximum to 1600:

```
DSLAM(config)# atm oam max-limit 1600
```

Configure the Interface-Level OAM

To enable OAM operations an interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**interface atm** *slot/port*[*.vpt#*]	Select the interface to be configured.
2	**atm oam** [*vpi* [*vci*] | **interface atm** *slot/port*[*.vpt#*]] [**ais**] [**end-loopback**] [**rdi**] [**seg-loopback**]	Configure interface OAM operations.
3	**atm oam** *vpi* [*vci*] **loopback-timer** *tx-timer-value*	Configure the OAM loopback transmit timer.

Note The OAM loopback command is hierarchial. You must first enable OAM at the global level then at the interface level before you can enable it on specific VCs.

Examples

This example shows how to enable OAM AIS end loopback on interface 0/1:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm oam ais end-loopback
% OAM: Interface level end to end loopback is enabled

% OAM: Interface level ais is enabled
```

The next example shows how to enable interface 0/1, VPI = 50, VCI = 100 to allow OAM AIS at the end and loopback:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm oam 50 100 ais end-loopback
% OAM: Connection level end to end loopback is not enabled

% OAM: Connection level ais is not enabled
```

Enable or disable OAM AIS, RDI, and loopback operations respective to a specified connection.

Note You can use only VPI values to turn on OAM operations on VP connections.

In interface configuration command mode, you can enable or disable OAM operations on existing connections on different interfaces by specifying **interface atm** *slot/port*. The third example enables OAM AIS flows at interface 0/1 level:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm oam ais
% OAM: Interface level ais is enabled
```

To view the result of this action, use the **show atm vc interface** command. This example displays the output you get for the ATM VC interface a0/1, with VPI = 0 and VCI = 500:

```
DSLAM# show atm vc interface a0/1 0 500

Interface: ATM0/1, Type: suni_dual
VPI = 0  VCI = 500
Status: UP
Time-since-last-status-change: 01:35:46
Connection-type: PVC
Cast-type: point-to-point
Packet-discard-option: disabled
Usage-Parameter-Control (UPC): pass
Number of OAM-configured connections: 6
OAM-configuration: Seg-loopback-on End-to-end-loopback-on
OAM-states:  OAM-Up End-to-end-loopback-up
OAM-Loopback-Tx-Interval: 5
Cross-connect-interface: ATM0/0, Type: ATM Swi/Proc
Cross-connect-VPI = 0
Cross-connect-VCI = 500
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Encapsulation: AAL5SNAP
Rx cells: 1220, Tx cells: 0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

Checking the ATM Connection

To check ATM connection reachability and network connectivity, use the EXEC command:

Step	Command	Task
1	**configure** [**terminal**]	At the privileged EXEC prompt, enter configuration mode from the terminal.
2	**interface atm** *slot/port*[*.vpt#*]	Select the interface.
3	**ping atm interface atm** *slot/port vpi vci* [**atm-prefix** *prefix*] | [**end-loopback**] | [**ip-address** *address*] | [**seg-loopback**]	Check the connection.

You can use either an IP address or an ATM address prefix as a ping destination. You can also ping a neighbor switch or DSLAM by selecting the segment loopback option. In privileged EXEC mode, you can select various other parameters such as repeat count and timeout values.

Examples

This example shows the **ping** command used in normal mode to check a VCC with a segment loopback signal:

```
DSLAM# ping atm interface atm 0/1 50 100 seg-loopback

Type escape sequence to abort.
Sending Seg-Loopback 5, 53-byte OAM Echoes to a neighbor, timeout is 5 seconds:
.....
Success rate is 0 percent (0/5)
```

This example shows the **ping** command used in extended mode to check a VCC with a segment loopback signal:

```
DSLAM# ping
Protocol [ip]: atm
Interface [card/sub-card/port]: 0/1
VPI [0]: 0
VCI [0]: 16
Send OAM-Segment-Loopback ? [no]:
Target IP address:
Target NSAP Prefix:
Repeat count [5]:
Timeout in seconds [5]:
Type escape sequence to abort.
Sending end-Loopback 5, 53-byte OAM Echoes to a connection end point, timeout is
 5 seconds:
.....
Success rate is 0 percent (0/5)
```

Note If you skip both destination IP address and the ATM prefix fields, then extended ping considers its neighbor switch as its destination and a segment loopback OAM cell.

To display ATM statistics for each active port on the DSLAM, use the command **show atm vc** . For example,

```
DSLAM> show atm vc int atm 13/2 11
```

For each VC/VP the following displays:

- Total Received Cells

- Total Dropped Cells

To display port-specific ATM statistics, use the command **show atm interface**. For example,

```
DSLAM> show atm int 13/2
```

For each port the following displays:

- Total Tx and Rx Cells
- Dropped cells due to HEC error
- Output port queue level (watermark)
- Aggregate Counters
- Discards due to invalid address
- Input and output queue overflow
- Exceeded contract violations (policing violations)
- ATM Diagnostics

Displaying the OAM Configuration

To display the OAM configuration, use the EXEC command:

Command	Task
show running-config	Display the OAM configuration.

Example

The OAM configuration is displayed in this example:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!
version XX.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
boot system flash slot0:rhino/6260-wi-m_1.083.bin.Z
!
ip rcmd remote-username doug
atm oam max-limit 1600
atm over-subscription-factor 16
atm service-category-limit cbr 3000
atm qos uni3-default  cbr max-cell-loss-ratio 12
atm lecs-address 47.0091.0000.0000.0000.0000.0000.0000.0000.0000.00
atm address 47.0091.8100.0000.0060.3e5a.db01.0060.3e5a.db01.00
!
interface ATM0/0
 no keepalive
 map-group atm-1
 no atm auto-configuration
```

```
 no atm address-registration
 no atm ilmi-enable
 no atm ilmi-lecs-implied
 atm iisp side user
 atm pvp 99
 atm oam 0 5 seg-loopback  end-loopback  rdi
 atm oam 0 16 seg-loopback  end-loopback  rdi
 atm oam 0 18 seg-loopback  end-loopback  rdi
!
interface ATM0/0.99 point-to-point
 no atm auto-configuration
 no atm address-registration
 no atm ilmi-enable
 no atm ilmi-lecs-implied
 atm maxvp-number 0
 atm oam 99 5 end-loopback  rdi
 atm oam 99 16 end-loopback  rdi
 atm oam 99 18 end-loopback  rdi
!
interface ATM0/1
 no keepalive
 --More--

<information deleted>
```

Configuring Digital Subscriber Lines

This chapter describes how to configure Cisco Digital Subscriber Line Access Multiplexers (DSLAMs) with NI-2 for digital subscriber line (DSL) service. The chapter contains the following sections:

Configuring Line Card Elements

The following sections discuss configuring ports and slots on line cards.

Enabling and Disabling a Port

This section describes how to enable or disable a port.

To enable a port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Go to the interface configuration mode and specify the port you want to enable.
3.	DSLAM(config-if)#**no shutdown**	Enable the specified port.

To disable a port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Go to the interface configuration mode and specify the port you want to disable.
3.	DSLAM(config-if)#**shutdown**	Disable the specified port.

Example

This example enables port 20 on slot 0 and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#interface atm 20/0
DSLAM(config-if)#no shutdown
DSLAM(config-if)#end
DSLAM#show dsl interface atm 20/0
```

```
Port Status:
   Subscriber Name:          Circuit ID:
   IOS admin: UP     oper: UP     Card status: Present
   Last Change: 36352 days, 13 hrs, 51 min, 47 sec No. of changes: 0
   Line Status: TRAINED
   Test Mode:   NONE

ADSL Chipset Self-Test: NONE

CO Modem Firmware Version: 0x1319BE02
.
.
.
```

Note The admin status is modified by the **shutdown** and **no shutdown** commands. The oper (operational) status is a function of the ATM switch fabric and the DSL line state.

Assigning Port Names

This section describes how to assign a name to a DSL subscriber port. The name may contain up to 64 printable characters. Alphanumerics and most special characters (underscores, hyphens, and ampersands, for example) are allowed. Spaces and quotes are not allowed.

To assign a name to a DSL subscriber port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Specify the slot and port.
3.	DSLAM(config-if)#**dsl subscriber** *name*	Assign *name* to the port.

Example

In this exmple, the name “curley” is assigned to slot 9, port 2.

```
DSLAM#configure terminal
DSLAM(config)#interface atm 9/2
DSLAM(config-if)#dsl subscriber curley
```

Assigning Circuit IDs

This section describes how to assign an identifier to a DSL circuit. The circuit ID may contain up to 32 printable characters. Alphanumerics and most special characters (underscores, hyphens, and ampersands, for example) are allowed. Spaces and quotes are not allowed.

To assign an identifier to a DSL circuit, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.

Step	Command	Task
2.	DSLAM(config)#**interface atm** *slot/port*	Specify the slot and port.
3.	DSLAM(config-if)#**dsl circuit** *circuit-id*	Assign *circuit-id* to the port.

Example

In this example, the circuit ID 341 is assigned to slot 9, port 2.

```
DSLAM#configure terminal
DSLAM(config)#interface atm 9/2
DSLAM(config-if)#dsl circuit 341
```

Displaying Debugging Information for a Port

This section describes how to display debugging information for a port.

To display debugging information for a port, follow this step:

Step	Command	Task
1.	DSLAM#**show controller atm** *slot/port*	Display debugging information for the selected port.

Command output includes

- Absolute SNR for each of the 32 upstream bins.
- Bit allocation for each of the 32 upstream bins.
- Configured and actual downstream transmit power boost. Configured powerboost displays in integer dB. Actual power boost displays in dB to one decimal place (0.1 dB) accuracy.
- Contents of the these chipset CMVs:
  - UOPT[7 : 0] (Upstream training options)
  - DOPT[7 : 0] (Downstream training options)
  - ADPT.downstream
  - ADPT.upstream
  - RATE.actual
  - RATE.maximum
  - CODE.upstream
  - CODE.downstream
  - INTL.upstream
  - INTL.downstream
  - DIAG.control
  - DIAG.flags_latched
  - PSDM.config
  - PSDM.actual

- OPTN.options
- OPTN.bitswap
- OPTN.utopia

Example

This example displays debugging information for slot 0, port 1:

```
DSLAM#show controller atm 20/1
ATM 20/1

Upstream SNR (in Tenths of dB)
      Sub Channel    SNR      Sub Channel    SNR
          0            0         16            0
          1            0         17            0
          2            0         18            0
          3            0         19            0
          4            0         20            0
          5            0         21            0
          6            0         22            0
          7            0         23            0
          8            0         24            0
          9            0         25            0
         10            0         26            0
         11            0         27            0
         12            0         28            0
         13            0         29            0
         14            0         30            0
         15            0         31            0

Upstream Bit Allocation
      Sub Channel    Bits Allocated      Sub Channel    Bits Allocated
          0                  0             16                  0
          1                  0             17                  0
          2                  0             18                  0
          3                  0             19                  0
          4                  0             20                  0
          5                  0             21                  0
          6                  0             22                  0
          7                  0             23                  0
          8                  0             24                  0
          9                  0             25                  0
         10                  8             26                  0
         11                  0             27                  0
         12                  0             28                  0
         13                  3             29                  0
         14                  0             30                  0
         15                  0             31                  0

Upstream TX Gain (in Tenths of dB)
      Sub Channel        TX Gain         Sub Channel        TX Gain
          0                  0             16                  0
          1                  0             17                  0
          2                  0             18                  0
          3                  0             19                  0
          4                  0             20                  0
          5                  0             21                  0
          6                  0             22                  0
          7                  0             23                  0
          8                  0             24                  0
          9                  0             25                  0
         10                  0             26                  0
```

```
         11                  0              27                   0
         12                  0              28                   0
         13                  0              29                   0
         14                  0              30                   0
         15                  0              31                   0

Downstream Bit Allocation
   0   16   32   48   64   80   96  112  128  144  160  176  192  208  224  240
-------------------------------------------------------------------------------
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    2    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

Downstream TX Gain (in Tenths of dB)
   0   16   32   48   64   80   96  112  128  144  160  176  192  208  224  240
-------------------------------------------------------------------------------
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

Configuring a Slot

To configure a slot for a specific card type, use these commands:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#slot` *slot# cardtype*	Configure the *slot#* to the desired *cardtype*.

The slot number range varies by platform; the maximum range is 1 to 38. These card types are available:

- ATUC-1-4DMT—4-port DMT card
- ATUC-1-4DMT-I—4-port DMT over ISDN card
- ATUC-4FLEXICAP—4-port Flexi card configured as CAP
- ATUC-4FLEXIDMT—4-port Flexi card configured as DMT
- ITUC-1-8IDSL—8-port IDSL card
- STUC-4-2B1Q-DIR-1—4-port SDSL card
- ATUC-8-DMT-1-H—8-port DMT OSP card
- STUC-8-TCPAM—G.SHDSL card

Note Some line cards do not function in all NI-2 DSLAM systems. For example, the Cisco 6100 system supports only a dual-port CAP ATU-C line card. Consult the hardware documentation for your DSLAM to determine which line cards it supports.

Example

This example configures slot 12 for a 4-port SDSL card and displays the hardware associated with the slot.

```
DSLAM#configure terminal
DSLAM(config)#slot 12 STUC-4-2B1Q-DIR-1
DSLAM#exit
DSLAM#show hardware slot 12

Slot 12: STUC-4-2B1Q-DIR-1

        Hardware Revision        : 2.0
        Part Number              : 800-07416-02
        Board Revision           : A0
        Deviation Number         : 0-0
        Fab Version              : 02
        PCB Serial Number        : FX900561224
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        CLEI Code                : VALITKFBAC
        Asset Identifier         :
        Platform features        : 48 79 AD 35 56 41 4C 49
                                   54 4B 46 42 41 43 BC C1
                                   7B 12 41 E8 E1 85 0C 41
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 40 01 D6 41 02 00 C0 46 03 20 00 1C F8 02
          0x10: 42 41 30 80 00 00 00 00 02 02 C1 8B 46 58 39 30
          0x20: 30 35 36 31 32 32 34 03 00 81 00 00 00 00 04 00
          0x30: C6 8A 56 41 4C 49 54 4B 46 42 41 43 CC 20 00 00
          0x40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          0x50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C9 18
          0x60: 48 79 AD 35 56 41 4C 49 54 4B 46 42 41 43 BC C1
          0x70: 7B 12 41 E8 E1 85 0C 41 FF FF FF FF FF FF FF FF
```

If the detected card type matches the slot provisioning for ATU-C and STU-C, the card type displays. The word “Missing” displays when a provisioned slot is empty.

"Mismatch" displays if the card type does not match the slot provisioning.

Note If you attempt to provision an empty slot, the major alarm "MODULE-MISSING" asserts.

Mixing Line Cards

The line coding used by the 4-port Flexi line card is spectrally incompatible with the line coding for both the 8-port IDSL line card and the 4-port SDSL (STU-C) line card. If you install spectrally incompatible cards in the same side of the chassis, the lines served by those cards can suffer reduced performance. For best performance in a chassis with a mixture of line card types, always install Flexi cards on one side of the chassis and install IDSL and SDSL cards on the opposite side.

Errors

Card mismatch error conditions occur under the following circumstances:

- If a line card of a different type is already installed in the named slot
- If you provision a slot for one type of card and insert another type of card into the named slot

Note You must provision an ATU-C FLEXI for CAP or DMT line coding before it will operate.

Using DSL Profiles

With the exception of a few dynamic operational modes, port configuration takes place through a configuration profile rather than by direct configuration. A profile is a named list of configuration parameters with a value assigned to each parameter. You can change the value of each parameter in the profile. To configure a subscriber, you need only attach the desired profile to that subscriber. When you change a parameter in a profile you change the value of that parameter on all ports using that profile. If you want to change a single port or a subset of ports, you can copy the profile, change the desired parameters, and then assign the new profile to the desired ports.

Note If you modify an existing profile, that change takes effect on every asymmetric digital subscriber line (ADSL) port linked to that profile.

This profile configuration approach is consistent with ADSL management information base (MIB) standards.

The DSLAM implementation uses the dynamic profile approach as opposed to the static profile approach. The dynamic profile approach supports a many-to-one correspondence between ports and profiles; that is, multiple ports can share the same profile but not vice versa. Also, with the dynamic approach, profiles are created and destroyed dynamically (with the exception of a special profile named "default"). Direct configuration of port parameters is not allowed.

Every port is attached to a special profile that is named "default." You can modify the "default" profile (but not delete it). This is useful when you want to modify one or two default parameters and apply this to every port in the system (rather than creating a new profile with minor changes and attaching this to every port in the system).

Note When you create a profile, it inherits all of the configuration settings of the special profile named "default" at the time of creation. If you subsequently modify the special profile "default," the changes do not propagate to profiles created using the original default profile.

Using profiles introduces a new command mode, profile mode. Use the command **dsl-profile** to enter profile mode. When you are in profile mode, changes you make to parameters affect only the profile you specify.

The following example sets the interleaved forward explicit congestion (FEC) check bytes for a profile named "test" to 6 upstream and 4 downstream. Other profiles do not change:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile test
DSLAM(cfg-dsl-profile)#dmt bitrate interleaved downstream 4 upstream 6
```

Creating, Modifying, or Deleting a Profile

This section describes how to create or delete a profile, and how to select a profile for modification.

To create a profile, or to select a profile for modification, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Create a profile named *profile-name*, or select an existing profile named *profile-name* for modification.

To delete a profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**no dsl-profile** *profile-name*	Deleted *profile-name*.

Note You can modify the default profile, but you cannot delete it.

Examples

The following example creates a DSL profile named "fast2." After you execute these steps, you can modify the parameters for this profile:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile fast2
DSLAM(cfg-dsl-profile)#
```

This example modifies the default profile and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#alarms
DSLAM(cfg-dsl-profile)#exit
DSLAM(config)#exit
DSLAM#show dsl profile default

Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
.
.
.
```

Copying a Profile

To copy a profile to an identical profile with a different name, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-copy-profile** [**force**] **source** *source-profile* **destination** *new-profile*	Copy the profile named *source-profile* to a profile named *new-profile*

If the destination profile indicated in this command does not exist, **dsl-copy-profile** creates it. The command then copies all non-default configurations defined for the source profile to the destination profile.

Example

This example copies the default profile to a profile named “fast” and displays the results. If “fast” does not exist, the command creates it. Use the command **show dsl profile** to confirm the existence and parameters for the new profile:

```
DSLAM#configure terminal
DSLAM(config)#dsl-copy-profile force source default destination fast
DSLAM(config)#exit
DSLAM#show dsl profile fast
dsl profile fast:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled
DMT profile parameters
      Maximum Bitrates:
         Interleave Path:  downstream:  8032/kbs,   upstream:   480/kbs
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled

 SDSL profile parameters
.
.
.
```

Attaching or Detaching a Profile

This section describes how to attach or detach a profile to or from a slot or port.

To attach a profile from a slot or port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Go to the interface configuration mode and specify the *slot/port* to which you want to attach the profile.
3.	DSLAM(config-if)#**dsl profile** *profile-name*	Attach *profile-name* to the slot/port.

To detach a profile from a slot or port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Go to the interface configuration mode and specify the *slot/port* from which you want to detach the profile.
3.	DSLAM(config-if)#**no dsl profile** *profile-name*	Detach *profile-name* from the specified slot/port.

Example

This example attaches the profile "test1" to slot 20, port 1, and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#interface atm 20/1
DSLAM(config-if)#dsl profile test1
DSLAM(config-if)#exit
DSLAM(config)#exit
DSLAM#show dsl interface atm 20/1
Port Status:
   Subscriber Name:          Circuit ID:
   IOS admin: UP     oper: UP     Card status: Present
   Last Change: 36352 days, 13 hrs, 51 min, 47 sec No. of changes: 0
   Line Status: TRAINED
   Test Mode:   NONE

ADSL Chipset Self-Test: NONE

CO Modem Firmware Version: O.21

Configured:
      DMT Profile Name: fast
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled
.
.
.
```

Displaying a Profile

To display a profile and all the ports currently connected to it, follow this step:

Step	Command	Task
1.	DSLAM#**show dsl profile** *profile-name*	Display a profile and all the ports currently connected to it.

Note If you omit the *profile-name* argument, this command displays profile information for all existing DSL profiles.

Example

This example displays the profile “fast”:

```
DSLAM#show dsl profile fast

dsl profile fast:
Link Traps Enabled: NO
      Alarms Enabled: YES
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled

 SDSL profile parameters
.
.
.
```

Displaying DSL Profiles

To display all nondefault settings for each currently defined DMT profile, including the default profile, follow this step:

Step	Command	Task
1.	DSLAM#**show running-config**	Display all nondefault settings for each currently defined DMT profile, including the default DMT profile.

Example

This example shows how to display a running configuration:

```
DSLAM#show running-config
Current configuration : 12125 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
!
hostname DSLAM
!
boot system flash:ni2-dsl-mz.v121_7_da.20010416
slot 1 ATUC-1-4DMT
slot 2 ATUC-1-4DMT
slot 3 ATUC-1-4DMT
slot 4 ATUC-1-4DMT
slot 5 ATUC-1-4DMT
slot 6 ATUC-1-4DMT
slot 7 ATUC-1-4DMT
slot 8 ATUC-1-4DMT
slot 9 ATUC-4FLEXIDMT
slot 10 NI-2-DS3-T1E1
slot 12 STUC-4-2B1Q-DIR-1
slot 13 ATUC-4FLEXIDMT
slot 14 STUC-4-2B1Q-DIR-1
slot 15 STUC-4-2B1Q-DIR-1
slot 16 STUC-4-2B1Q-DIR-1
slot 17 STUC-4-2B1Q-DIR-1
slot 18 ATUC-1-DMT8
slot 19 ATUC-1-4DMT
slot 20 ATUC-1-DMT8
slot 21 ATUC-1-4DMT
slot 22 STUC-4-2B1Q-DIR-1
slot 23 ATUC-1-4DMT
slot 24 ATUC-1-4DMT
slot 25 ATUC-1-4DMT
slot 26 ATUC-1-4DMT
slot 27 ATUC-4FLEXIDMT
slot 28 ATUC-1-4DMT
slot 29 ATUC-1-DMT8
slot 30 ATUC-1-4DMT
slot 31 STUC-8-TCPAM
slot 32 ATUC-1-4DMT-I
no logging console
enable password cisco
!
!
!
!
!
!
dsl-profile default
!
dsl-profile brent
 dmt overhead-framing mode1
 dmt encoding trellis
 dmt margin downstream 4 upstream 4
 dmt bitrate maximum interleaved downstream 1024 upstream 384
!
dsl-profile fast
!
dsl-profile ADSL-Plus
 dmt bitrate maximum interleaved downstream 512 upstream 160
!
dsl-profile residencial
 dmt codeword-size downstream 16 upstream 16
 dmt bitrate maximum interleaved downstream 256 upstream 160
!
!
atm oam max-limit 1600
no atm oam intercept end-to-end
```

```
atm address 47.0091.8100.0000.0001.64ff.a980.0001.64ff.a980.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
atm ni2-switch trunk ATM0/IMA0
!
icm size 4194304
!
!
interface ATM0/0
 no ip address
 atm maxvp-number 0
 atm maxvc-number 4096
 atm maxvci-bits 12
!
interface Ethernet0/0
 ip address 172.21.186.145 255.255.255.192
!
interface ATM0/2
 no ip address
 no atm ilmi-keepalive
 atm oam 0 5 seg-loopback
 atm oam 0 16 seg-loopback
 clock source loop-timed
 framing crc4
 lbo short gain10
 ima-group 0
!
.
.
.
```

Setting DSL Profile Parameters

This section describes the various parameters that can be set within a DSL profile.

Enabling and Disabling Alarms

You can enable and disable alarms for a selected DSL profile using a single command. The alarms apply to these event classes:

- Near End LOS (loss of signal)
- Near End LOCD (loss of cell delineation)
- Near End LOF (loss of frame)
- ATU-C DMT port failure
- Up and/or downstream bit rate not above minimum bit rate

DSL alarms are disabled by default.

To enable DSL alarms, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify a profile.
3.	DSLAM(cfg-dsl-profile)#**alarms**	Enable alarms for that profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To disable DSL alarms, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify a profile.
3.	DSLAM(cfg-dsl-profile)#**no alarms**	Disable alarms for that profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Example

This example enables alarms for the default profile and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#alarms
DSLAM(cfg-dsl-profile)#end
DSLAM#show dsl profile default

dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: YES
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path: downstream:  640/kbs,   upstream:  128/kbs
      Minimum Bitrates:
         Interleave Path: downstream:    0/kbs,   upstream:    0/kbs
         .
         .
         .
```

Enabling and Disabling Payload Scrambling

This section describes how to enable and disable cell payload scrambling on a DMT subscriber port. Payload scrambling is enabled by default.

To enable payload scrambling, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify the *profile-name* for which you want to enable payload scrambling.
3.	DSLAM(cfg-dsl-profile)#**payload-scrambling**	Enable payload scrambling.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To disable trellis coding, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify the *profile-name* for which you want to disable payload scrambling.
3.	DSLAM(cfg-dsl-profile)#**no payload-scrambling**	Disable payload scrambling.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

The two ends of a connection must have the same payload scrambling value—that is, payload scrambling must be enabled at both ends or disabled at both ends. The line does not train if payload scrambling is enabled at one end and disabled at the other end.

Enabling or disabling payload scrambling does not cause the port to retrain.

Setting CAP Upstream and Downstream Baud Rates

This section describes how to configure upstream and downstream baud rate margins for ATU-C CAP, and ATU-C FLEXI CAP interfaces.

Cisco IOS supports provisioning additional baud rates for interface line codes. The following rules apply:

- Valid rate, Cisco IOS selects a rate less than or equal to the rate that you specified.
- Invalid rate, Cisco IOS modifies the rate to the closest available rate that is less than or equal to the rate that you specified.

In addition to the existing upstream 136 kilobaud rate, Cisco IOS also supports an upstream 17 kilobaud rate and an upstream 68 kilobaud rate. You can independently enable or disable the new baud rates.

The following list contains the valid upstream/downstream pairs within the available rates:

- An upstream rate of 17 kilobaud is valid only with a downstream rate of 136 kilobaud.
- An upstream rate of 68 kilobaud is valid only with a downstream rate of 136 kilobaud or a downstream rate of 340 kilobaud.

- All other combinations are valid.

Table 7-1 and Table 7-2 show the upstream and downstream baud rates and their corresponding bit rates for the ATU-C CAP and ATU-C FLEXI CAP interfaces.

Table 7-1 ATU-C CAP and ATU-C FLEXI CAP Upstream Baud Rates and Corresponding Bit Rates

Module	Upstream Baud Rate	Upstream Bit Rate (kbps)
ATU-C CAP/ ATU-C FLEXI CAP	136 kilobaud	1088, 952, 816, 680, 544, 408, 272 91
	68 kilobaud	544, 476, 408, 340, 272, 204, 136, 46
	17 kilobaud	136, 119, 102, 85, 68, 51, 34, 12

Table 7-2 ATU-C CAP and ATU-C FLEXI CAP Downstream Baud Rates and Corresponding Bit Rates

Module	Downstream Baud Rate	Downstream Bit Rate (kbps)
ATU-C CAP/ ATU-C FLEXI CAP	952 kilobaud	7168, 6272, 4480, 2688
	680 kilobaud	5120, 4480, 3200, 1920
	340 kilobaud	2560, 2240, 1920, 1600, 1280, 960, 640
	136 kilobaud—RS[1] enabled	1024, 896, 768, 640, 512, 384, 256
	136 kilobaud—RS disabled	1088, 952, 816, 680, 544, 408, 272

1. Reed-Solomon coding—long/short interleave

The following information applies to Table 7-1 and Table 7-2:

- Enabling 17 kilobaud upstream and 68 kilobaud upstream rates are not mutually exclusive.
- The valid upstream rates are the union of the common rates (136 kilobaud upstream) and the bit rates corresponding to the new bauds (17 kilobaud upstream and 68 kilobaud upstream).
- If a given upstream rate appears in more than one selected baud rate list, the higher baud rate applies.

To enable baud rates, follow these steps:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#dsl-profile` *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to enable baud rates.
3.	`DSLAM(cfg-dsl-profile)#cap baud` {`downstream` *baudrate* | `upstream` { *baudrate* | *baudrate*}}	Enable one or more baud rates for the designated CAP profile.
4.	`DSLAM(cfg-dsl-profile)#end`	Exit from profile configuration mode.

To disable baud rates, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to disable baud rates.
3.	DSLAM(cfg-dsl-profile)#**no cap baud** {**downstream** *baudrate* | **upstream** { *baudrate* | *baudrate* }}	Disable one or more baud rates for the specified CAP profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Setting Upstream and Downstream Bit Rates

This section describes how to configure upstream and downstream bit rates for ATU-C CAP, and ATU-C FLEXI CAP, DMT, and STU-C interfaces.

Setting Bit Rate Parameters for ATU-C CAP Interfaces

To set the downstream and upstream minimum or maximum bit rates for a CAP interface, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the bit rate.
3.	DSLAM(cfg-dsl-profile)#**cap bitrate** {**minimum** | **maximum**} **downstream** *int* **upstream** *int*	Set the bitrate for downstream and upstream for the CAP interface for this profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To return the downstream and upstream bit rates for a CAP interface to their default values, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the default bit rate.
3.	DSLAM(cfg-dsl-profile)#**no cap bitrate** {**minimum** | **maximum** } **downstream** *int* **upstream** *int*	Set this profile to the default bit rate.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Defaults

The following are the default minimum and maximum downstream and upstream bit rates for the ATU-C CAP interface.

Value Type	Default
Minimum downstream	0 kbps
Minimum upstream	0 kbps
Maximum downstream	640 kbps
Maximum upstream	91 kbps

The alarm subsystem uses the minimum bit rate settings. The Cisco IOS asserts an alarm if the line card trains at a rate below the configured minimum bit rate.

Examples

In this example, the command sets the maximum downstream and upstream bit rates to 7168 kbps, and 1088 kbps respectively:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#cap bitrate maximum downstream 7168 upstream 1088
DSLAM(cfg-dsl-profile)#end
```

In this example, the command sets the maximum downstream and upstream bit rates to the default values for that particular interface. In this case, it is a quad port ATU-C FLEXI CAP.

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#cap bitrate maximum downstream 5150 upstream 880
DSLAM(cfg-dsl-profile)#end
```

Setting Bit Rate Parameters for DMT Interfaces

To set the maximum allowed bit rate for interleaved-path DMT parameters for a specific profile, follow these stesps:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#dsl-profile` *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the maximum allowed bit rate for interleaved-path DMT profile parameters.

Step	Command	Task
3.	DSLAM(cfg-dsl-profile)#**dmt bitrate max interleaved-path downstream** *dmt-bitrate* **upstream** *dmt-bitrate*	Set the maximum allowed downstream and upstream bit rate for interleaved-path DMT profile parameters to *dmt-bitrate*.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To set the minimum allowed bit rate for interleaved-path DMT parameters for a specific profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the minimum allowed bit rate for interleaved-path DMT profile parameters
3.	DSLAM(cfg-dsl-profile)#**dmt bitrate min interleaved-path downstream** *dmt-bitrate* **upstream** *dmt-bitrate*	Set the maximum allowed downstream and upstream bit rate for interleaved-path DMT profile parameters to *dmt-bitrate*.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

The software does not send minimum bit rate settings to the line card. The Cisco IOS software uses these settings locally to determine if a line rate alarm should be set for a port.

Setting the DMT bit rate to 0 disables the associated minimum DMT bit rate alarm.

Table 7-3 lists the allowable ranges and default values for DMT bit rate.

Table 7-3 Allowable Ranges and Default Values for DMT Bit Rates

Configuration Parameter	Data Path	Downstream			Upstream		
		Aggregate Range (kbps)	Path Range (kbps)	Path Default (kbps)	Aggregate Range (kbps)	Path Range (kbps)	Path Default (kbps)
dmt bitrate max	Interleaved	8032 to 32	8032 to 32	640	864 to 32	864 to 0	128
dmt bitrate min	Interleaved	8032 to 32	8032 to 0	0	864 to 0	864 to 0	0

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

Example

This example sets the maximum interleaved-path bit rate of the default profile to 640 kbps downstream, and 128 kbps upstream and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
```

```
DSLAM(cfg-dsl-profile)#dmt bitrate interleaved-path downstream 640 upstream 128
DSLAM(cfg-dsl-profile)#end
DSLAM#show dsl-profile

dsl profile default:
      Alarms Enabled: NO

 DMT profile parameters
      Maximum Bitrates:
Interleave Path:  downstream:   640/kbs,   upstream:   128/kbs
      Minimum Bitrates:
         Interleave Path:  downstream:     0/kbs,   upstream:     0/kbs
      Margin:              downstream:     3 db,    upstream:     3 db
      Interleave Delay:    downstream: 16000 usecs, upstream: 16000 usecs
      FEC Redundancy Bytes:
         Interleave Path:  downstream:    16,       upstream:    16
      R-S Codeword Size:   downstream:  auto,       upstream:  auto
      Trellis Coding:         Enabled
      Overhead Framing:       Mode 1
      Bit-Swap:               Enabled
      Bit-Swap From Margin:      3 dB
      Bit-Swap To Margin:        3 dB
      Operating Mode:         Automatic
      Training Mode:          Standard

 SDSL profile parameters
.
.
.
```

Setting Bit Rate Parameters for STU-C Interfaces

To set the bit rate for STU-C parameters for a profile, follow these steps:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#dsl-profile` *profile-name*	Go to the profile configuration mode, specifying the *profile-name* for which you want to set the maximum allowed bit rate.
3.	`DSLAM(cfg-dsl-profile)#sdsl bitrate` *bitrate*	Set the downstream and upstream bit rates for the profile. The STU-C downstream and upstream bit rates are identical. The loop characteristics determine the achievable rate.
4.	`DSLAM(cfg-dsl-profile)#end`	Exit from profile configuration mode.

Example

In this example, the command sets the bit rate of the default profile to 528 kbps downstream and upstream:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#sdsl bitrate 528
```

The Cisco IOS software does not send minimum bit rate settings to the STU-C line card. The software uses the settings locally to determine if a line rate alarm should be set for a port.

The following allowable STU-C bit rate ranges occur in units of kbps:

1168

1040

784

528

400

272

144

Caution The **sdsl bitrate** *bitrate* command causes the port to retrain when you change the parameter.

Setting a parameter to its current value does not cause a retrain. If a port is training when you change the parameter, the port untrains and retrains to the new parameter value.

Setting Signal-to-Noise Ratio Margins

This section describes how to set signal-to-noise ratio (SNR) margins for both downstream and upstream traffic for ATU-C CAP, ATU-C Flexi CAP, ATU-C Flexi DMT and 4DMT interfaces. The higher the SNR margin the more protection there is against data corruption. The higher the SNR margin the lower the data rate a given loop can support.

ATU-C CAP and ATU-C FLEXI CAP Interfaces

Use the following profile configuration commands set the SNR value for a selected ATU-C CAP or ATU-C Flexi CAP profile:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#dsl-profile` *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set SNR margins.
3.	`DSLAM(cfg-dsl-profile)#cap margin downstream` *0-12* `upstream` *0-12*	Set the SNR downstream and upstream margins to integers 0 through 12.
4.	`DSLAM(cfg-dsl-profile)#end`	Exit from profile configuration mode.

To set the SNR margin values for an ATU-C CAP interface to the default values of 6 dB in both directions, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set SNR margins.
3.	DSLAM(cfg-dsl-profile)#**no cap margin {downstream | upstream}**	Set the SNR downstream or upstream margins to the default value (6 dB).
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Example

In this example, the command sets the SNR margin at 8 dB downstream and 5 dB upstream for the DSL profile “issis”:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#cap margin downstream 8 upstream 5
DSLAM(cfg-dsl-profile)#end
```

ATU-C 4DMT Interface

The range of DMT margin values is 0 to 15 dB in each direction. The default value for each direction is 6 dB.

To set SNR margins for a 4DMT interface, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set SNR margins.
3.	DSLAM(cfg-dsl-profile)#**dmt margin downstream** *dmt-margin* **upstream** *dmt-margin*	Set the SNR downstream and upstream margins to *dmt-margin*.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

Example

This example sets the SNR margins of the default profile to 6 dB upstream and 6 dB downstream and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#dmt margin downstream 6 upstream 6
DSLAM(cfg-dsl-profile)#end
DSLAM#show running-config

Building configuration...

Current configuration:
!
!
version XX.X
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname DSLAM
!
slot 1 atuc-1-4dmt
.
.
.
slot 32 atuc-1-4dmt
enable password lab
!
!
dsl-profile default
!
dsl-profile fast
 dmt training-mode quick
dmt margin downstream 6 upstream 6
 dmt bitrate maximum interleaved downstream 8032 upstream 480
network-clock-select 1 ATM0/1
network-clock-select 2 system
.
.
.
```

Setting the Interleaving Delay

This section describes how to set the interleaving delay for both the upstream and downstream traffic for DMT and CAP interfaces.

If possible, the DSLAM sets the actual interleaving delays to match the values configured in the profile. However, depending upon the bit rate to which the port finally trains, some settings of interleaving delay may not be achievable. In this case, the DSLAM chooses an actual interleaving delay that is closest (numerically) to the configured interleaving delay. Table 7-4 lists the values of interleaving delay that are achievable for all bit rates.

DMT Interfaces

Interleaving delay helps protect against impulse noise and clipping, but adds delay, which may not be tolerable for some applications.

The allowable values for configured interleaved delay are 0, 500, 1000, 2000, 4000, 8000, and 16000 microseconds. The default interleaved delay (the value assigned when a DSL profile is created) is 16000 microseconds (that is, 16 msecs) for both upstream and downstream directions.

Table 7-4 Achievable Combinations of Interleaving Delay and Symbols Per R-S Codeword for Different Bit Rate Ranges

Bit Rate Range (kbps)	Symbols per R-S Codeword Allowed	Interleaving Delay Allowed (microseconds)
8032 to 3616	1	0, 500, 1000, 2000, 8000, 16000
3584 to 3168	1 or 2	0, 500, 1000, 2000, 8000, 16000 **Note** A value of 500 is allowed only when symbols per codeword = 1.
3136 to 1760	2	0, 1000, 2000, 8000, 16000
1728 to 1568	2 or 4	0, 1000, 2000, 4000, 8000, 16000 **Note** A value 1000 is allowed only when symbols per codeword = 2. A value of 4000 is allowed only when symbols per codeword = 4.
1536 to 832	4	0, 2000, 4000, 8000, 16000
800 to 768	4 or 8	0, 2000, 4000, 8000, 16000 **Note** A value of 2000 is allowed only when symbols per codeword = 4.
736 to 384	8	0, 4000, 8000, 16000
352 to 0	16	0, 8000, 16000

To set upstream and downstream interleaved delay for a specific DMT profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the interleaving delay.
3.	DSLAM(cfg-dsl-profile)#**dmt interleaving-delay downstream** *delay-in-usecs* **upstream** *delay-in-usecs*	Set the downstream and upstream interleaving delay times as *delay-in-usecs*.

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

Example

This example sets the interleaving delay of the profile named “fast” to 2000 usec downstream and 4000 usec upstream, and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile fast
DSLAM(cfg-dsl-profile)#dmt interleaving-delay downstream 2000 upstream 4000
DSLAM(cfg-dsl-profile)#exit
DSLAM(config)#exit
DSLAM#show dsl profile fast

dsl profile fast:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream:  2000 usecs,  upstream:  4000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
.
.
.
```

CAP Interfaces

Table 7-5 shows the amount of delay (in milliseconds) that results from various combinations of baud rate, constellation, and **cap interleaving-delay** settings (short or long) on a 4-port Flexi card configured for CAP. Interleaving delay is applied only in the downstream direction. Interleaving is not used on upstream traffic.

Table 7-5 Downstream Interleaving Delay

Constellation	Short or Long Delay	136 Kbaud	340 Kbaud	680 Kbaud	952 Kbaud
8	short	4.4 ms	4.4 ms	–	–
	long	49 ms	49 ms	–	–
16	short	3.0 ms	3.0 ms	3.0 ms	2.7 ms
	long	31 ms	31 ms	16 ms	11 ms
32	short	2.3 ms	2.3 ms	–	–
	long	24 ms	24 ms	–	–
64	short	1.9 ms	1.9 ms	1.8 ms	1.7 ms
	long	19 ms	19 ms	9.6 ms	6.8 ms
128	short	1.6 ms	1.6 ms	–	–
	long	16 ms	16 ms	–	–
256	short	1.4 ms	1.4 ms	1.4 ms	1.2 ms
	long	14 ms	14 ms	6.8 ms	5.0 ms
256 uncorrected	short	1.3 ms	1.3 ms	1.2 ms	1.0 ms
	long	12 ms	12 ms	6.0 ms	4.3 ms

You can choose the interleaving-delay option **none** only when 136k downstream baud rate is enabled. If you configure the interleaving-delay as **none** but the line card trains at a downstream bit rate that uses a baud rate that is other than 136k, the actual interleaving-delay value is **short**.

The following table shows the relationship between the interleaving-delay value chosen and the state of the Reed-Solomon error correction function.

Interleave Value	Reed-Solomon Relationship
Short	RS error correction on
Long	RS error correction on
None	RS error correction off

Note If you set interleaving delay to **none**, the subscriber's line may provide service at a higher bit rate than the one configured. This can happen because setting interleaving delay to **none** turns off Reed-Solomon error correction, and turning off error correction reduces the overhead on the line, leaving more bandwidth available to the subscriber.

To set the interleaving delay for a specific CAP profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the interleaving delay.
3.	DSLAM(cfg-dsl-profile)#**cap interleaving-delay** {**short** | **long** | **none**}	Set interleaving-delay for a designated CAP profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Return to priviledged EXEC mode.

To return the interleaved delay to its default (long) setting, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)# **dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the interleaving delay.
3.	DSLAM(cfg-dsl-profile)#**no cap interleaving-delay**	Set interleaving-delay to the default value (long) for a designated CAP profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Examples

This example shows how to set the interleaving-delay value to **none** for the profile named “issis”:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#cap interleaving-delay
DSLAM(cfg-dsl-profile)#end
```

This example shows how to set the default interleaving delay value for the profile named “issis”.

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#no cap interleaving-delay none
DSLAM(cfg-dsl-profile)#end
```

Setting the Number of Symbols Per Reed-Solomon Codeword

This section describes how to set the number of symbols per Reed-Solomon codeword. This information applies to DMT interfaces only.

The allowable values for configured symbols per codeword are 1, 2, 4, 8, 16, or auto. If you select auto (automatic), the line card chooses the optimum symbols per codeword based upon the bit rate to which the line trains. The optimum value keeps the ratio of user data to error correction bytes roughly constant. The default symbols per codeword setting (the value assigned when a DSL profile is created) is auto for both upstream and downstream directions.

If the symbols per codeword is set explicitly (any value other than auto), the DSLAM attempts to match the configured symbols per codeword. However, depending upon the bit rate to which the port finally trains, some settings of symbols per codeword may not be acheivable. When this occurs, the DSLAM chooses an actual symbols per codeword value that is closest (numerically) to the configured symbols per codeword. Table 7-6 lists the values of symbols per codeword that are allowable for various bit rate ranges.

Table 7-6 Symbols Per Codeword Values for Different Bit Rate Ranges

Bit Rate Range (kbps)	Symbols per R-S Codeword for Auto	Symbols per R-S Codeword Allowed
8032 to 3616	1	1
3584 to 3168	2	1 or 2
3136 to 1760	2	2
1728 to 1568	4	2 or 4
1536 to 832	4	4
800 to 768	8	4 or 8
736 to 384	8	8
352 to 0	16	16

When the training mode is set to quick the modem DSP automatically chooses the codeword size. The one exception is that if check bytes is set to zero and the training mode is quick, the codeword size is always one.

To set the number of symbols per Reed-Solomon codeword, follow these steps:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#dsl-profile default`	Go to the profile mode.
3.	`DSLAM(cfg-dsl-profile)#dmt codeword-size downstream {symbols | auto} upstream {symbols | auto}`	Set codeword size. The allowable values for codeword size (in symbols per R-S codeword) are 1, 2, 4, 8, 16, or auto.
4.	`DSLAM(cfg-dsl-profile)#end`	Exit from profile configuration mode.

If you set the codeword size to **auto**, the number of symbols per codeword will depend upon the actual DMT bit rate. The default codeword size is auto.

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

Example

This example sets the number of symbols per Reed-Solomon codeword to 8 upstream and 16 downstream and displays the results:

```
DSLAM# configure terminal
DSLAM(config)# dsl-profile default
DSLAM(cfg-dsl-profile)# dmt codeword-size downstream 16 upstream 8
DSLAM(cfg-dsl-profile)#end
DSLAM# show dsl profile default

dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:    16,        upstream:     8
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
.
.
.
```

Setting FEC Check (Redundancy) Bytes

This section describes how to set upstream and downstream interleaved FEC check (redundancy) bytes per Reed-Solomon (R-S) codeword for a specific profile for DMT interfaces. The higher the check bytes setting, the better the error correction, but the check bytes subtract from user bytes.

The configured number of FEC check bytes must be an even number in the range 0 to 16 inclusive. The default (the value assigned when a DSL profile is created) is 16 check bytes for both the upstream and downstream directions.

If possible, the DSLAM sets the actual number of FEC check bytes to match the value configured in the profile. However, depending upon the bit rate to which the port finally trains, some settings of FEC check bytes may not be achievable. In this case, the DSLAM chooses an actual number of FEC check bytes that is closest (numerically) to the configured number of FEC check bytes. Table 7-7 lists the values of FEC check bytes that are achievable for all bit rates.

Table 7-7 Achievable Combinations of FEC Check Bytes and Symbols Per Reed-Solomon Codeword for Different Bit Rate Ranges

Bit Rate Range (kbps)	Symbols per R-S Codeword Allowed	FEC Check Bytes Allowed
8032 to 3616	1	0, 2, 4, 6, 8, 10, 12, 14, 16
3584 to 3168	1 or 2	0, 2, 4, 6, 8, 10, 12, 14, 16

Table 7-7 Achievable Combinations of FEC Check Bytes and Symbols Per Reed-Solomon Codeword for Different Bit Rate Ranges

Bit Rate Range (kbps)	Symbols per R-S Codeword Allowed	FEC Check Bytes Allowed
3136 to 1760	2	0, 2, 4, 6, 8, 10, 12, 14, 16
1728 to 1568	2 or 4	0, 2, 4, 6, 8, 10, 12, 14, 16 **Note** Values of 2, 6, 10, or 14 are allowed only when symbols per R-S codeword = 2.
1536 to 832	4	0, 4, 8, 12, 16
800 to 768	4 or 8	0, 4, 8, 12, 16 **Note** Values of 4 or 12 are allowed only when symbols per R-S codeword = 4.
736 to 384	8	0, 8, 16
352 to 0	16	0, 16

To set upstream and downstream FEC check (redundancy) bytes for a specific profile, follow these steps:

Step	Command	Task
1.	`DSLAM#configure terminal`	Go to the global configuration mode.
2.	`DSLAM(config)#dsl-profile` *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set FEC check bytes.
3.	`DSLAM(cfg-dsl-profile)#dmt check-bytes interleaved downstream` *bytes* `upstream` *bytes*	Set the check bytes to the specified number of *bytes* downstream and *bytes* upstream.

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

It is normally desirable to keep the ratio of check bytes to user bytes roughly constant regardless of the bit rate. This requires you to change both the check bytes and the codeword size parameters.

When the training mode is set to quick, the DSLAM automatically chooses the check bytes value. However, if check bytes is set to zero and the training mode is quick, the system always uses a check bytes value of 0.

Example

This example sets the FEC check bytes for the default profile to 6 upstream and 4 downstream and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
```

```
DSLAM(cfg-dsl-profile)#dmt check-bytes interleaved downstream 4 upstream 6
DSLAM(cfg-dsl-profile)#end
DSLAM# show dsl profile default

dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
.
.
.
```

Enabling and Disabling Trellis Coding

This section describes how to enable or disable trellis coding.

To enable trellis coding, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify the *profile-name* for which you want to enable trellis coding.
3.	DSLAM(cfg-dsl-profile)#**dmt encoding trellis**	Enable trellis coding.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To disable trellis coding, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Specify the *profile-name* for which you want to disable trellis coding.
3.	DSLAM(cfg-dsl-profile)#**no dmt encoding trellis**	Disable trellis coding.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

The system can use trellis coding only if the profile enables it and the CPE supports trellis coding.

Example

This example turns off trellis encoding for the default profile and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#no dmt encoding trellis
DSLAM(cfg-dsl-profile)#end
DSLAM#show dsl profile
dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: YES
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:    16,        upstream:     8
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 2
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
.
.
.
```

Setting the Overhead Framing Mode

To set the overhead framing mode of a DMT profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the overhead framing mode.
3.	DSLAM(cfg-dsl-profile)#**dmt overhead-framing** {**mode1** | **mode2** | **mode3**}	Set the overhead framing mode.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

This command does not retrain the port when you change the parameter value.

If the actual framing mode used is the mode the ATU-C port requested, or if the ATU-R CPE does not support the ATU-C's choice, then the highest mode the ATU-R does support is used.

Example

This example sets the overhead framing mode in the default profile to mode2 and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#dmt overhead-framing mode2
DSLAM(cfg-dsl-profile)#end
DSLAM#show dsl profile

dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: YES
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:    16,        upstream:     8
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 2
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
```

Modifying the Operating Mode

To modify the operating mode of a DMT profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to modify the operating mode.
3.	DSLAM(cfg-dsl-profile)#**dmt operating-mode {auto | g992-1 | g992-2 | t1-413}**	Set an operating mode for the selected profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To set the operating mode of a DMT profile to the default mode, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to modify the operating mode.
3.	DSLAM(cfg-dsl-profile)#**no dmt operating-mode**	Force the operating mode to the default mode, auto.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

An ADSL line uses of one of these operating modes:

- **auto**—An ATU-C port that employs this operating mode automatically detects the capabilities of the ATU-R CPE and uses a startup sequence specified by either G.992.1, G.992.2, or T1.413-1998, or splitterless mode. Auto mode is the default for an ADSL line.
- **g992-1**—In this mode the line uses the G994.1 startup sequence. After startup, the line complies to G992.1 operation.
- **g992-2**—In this mode the line uses the G994.1 startup sequence. After startup, the line complies to G992.2 operation. (G992.2 is also known as G.lite.)
- **t1-413**—This mode forces the ATU-R CPE to use the T1.413-1998 startup sequence.

This command retrains the port if you change the parameter value. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter value.

Example

This example sets the operating mode of the default profile to splitterless and displays the results:

```
DSLAM# configure terminal
DSLAM(config)# dsl-profile default
```

```
DSLAM(cfg-dsl-profile)# dmt operating-mode splitterless
DSLAM# show dsl profile default

dsl profile default:
      Alarms Enabled: NO
DMT profile parameters
      Maximum Bitrates:
         Interleave Path:  downstream:   640/kbs,   upstream:   128/kbs
      Minimum Bitrates:
         Interleave Path:  downstream:     0/kbs,   upstream:     0/kbs
      Margin:              downstream:     3 db,    upstream:     3 db
      Interleave Delay:    downstream: 16000 usecs, upstream: 16000 usecs
      FEC Redundancy Bytes:
         Interleave Path:  downstream:    16,       upstream:    16
      R-S Codeword Size:   downstream:  auto,       upstream:  auto
      Trellis Coding:         Enabled
      Overhead Framing:       Mode 1
      Bit-Swap:               Enabled
      Bit-Swap From Margin:      3 dB
      Bit-Swap To Margin:        3 dB
 Operating Mode:         Splitterless
      Training Mode:          Standard
.
.
.
```

Modifying the Training Mode

To modify the training mode of a DMT profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to modify the training mode.
3.	DSLAM(cfg-dsl-profile)#**dmt training-mode** {**standard** | **quick**}	Modify the training mode. The choices are **standard** and **quick**.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

To set the training mode of a DMT profile to its default value, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to modify the training mode.

Step	Command	Task
3.	DSLAM(cfg-dsl-profile)#**no dmt training-mode**	Set the training mode to its default value.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

This object specifies the mode employed by the ATU-C port when it is training to an ATU-R CPE. There are two training modes:

- **Standard**—This mode uses the G.994.1 or T1.413-1998 initialization sequence depending on configuration. In standard training mode the ATU-C port trains with the modem once, and if the configured rates and settings are not obtainable, the line card reads the line quality and retrains, selecting the best available rates and settings. This mode allows more control over the DMT parameters.
- **Quick**—This mode is the default. It uses the extended exchange sequence for T1.413-1998 initialization or the G.994.1 initialization, depending on configuration. In quick training mode the modem DSP automatically determines the best available rate based on the parameters provided. The DSP may be forced to change some of the configuration settings based on the line characteristics. This training mode is faster than the standard mode.

This command does not retrain the port when you change the parameter value.

Example

This example sets the training mode of the default profile to quick and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#dmt training-mode quick
DSLAM(cfg-dsl-profile)#end
DSLAM#show dsl profile default

dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: YES
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:    16,        upstream:     8
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 2
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
```

```
 SDSL profile parameters
.
.
.
```

Setting DMT Margins for Bitswapping

This section describes how to set discrete multitone (DMT) margins for bitswapping.

To set the DMT margins for bitswapping, follow these steps:

Step	Command	Task
1	DSLAM#**configure terminal**	Go to the global command mode.
2	DSLAM(config)#**dsl-profile default**	Go to the profile mode.
3	DSLAM(cfg-dsl-profile)#**dmt bit-swap margin from** *dmt-margin* **to** *dmt-margin*	Set the bitswap margins.

This command does not retrain the port if you change the parameter value.

Example

This example sets the bitswap *from* DMT margin in the default profile to 3 and the bitswap *to* DMT margin to 3 and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile default
DSLAM(cfg-dsl-profile)#dmt bit-swap margin from 3 to 3
DSLAM(cfg-dsl-profile)#end
DSLAM#show dsl profile default

dsl profile default:
      Alarms Enabled:NO

 DMT profile parameters
      Maximum Bitrates:
         Interleave Path: downstream:  640/kbs,   upstream:  128/kbs
      Minimum Bitrates:
         Interleave Path: downstream:    0/kbs,   upstream:    0/kbs
      Margin:             downstream:    3 db,    upstream:    3 db
      Interleave Delay:   downstream:16000 usecs, upstream:16000 usecs
      FEC Redundancy Bytes:
         Interleave Path: downstream:   16,       upstream:   16
      R-S Codeword Size:  downstream: auto,       upstream: auto
      Trellis Coding:        Enabled
      Overhead Framing:      Mode 1
      Bit-Swap:              Enabled
 Bit-Swap From Margin:     3 dB
Bit-Swap To Margin:       3 dB
      Operating Mode:        Automatic
      Training Mode:         Standard

 SDSL profile parameters

 CAP profile parameters
```

Disabling Bitswapping

To disable bitswapping, follow these steps:

Step	Command	Task
1	DSLAM#**configure terminal**	Go to the global command mode.
2	DSLAM(config)#**dsl-profile default**	Go to the profile configuration mode.
3	DSLAM(cfg-dsl-profile)#**no dmt bit-swap**	Disable bitswapping.

This command does not retrain the port if you change the parameter value.

Example

This example disables bit-swapping for the default profile and displays the results:

```
DSLAM# configure terminal
DSLAM(config)# dsl-profile default
DSLAM(cfg-dsl-profile)# no dmt bit-swap
DSLAM(cfg-dsl-profile)#end
DSLAM# show dsl profile default

dsl profile default:
      Alarms Enabled: NO
DMT profile parameters
      Maximum Bitrates:
         Interleave Path:  downstream:   640/kbs,   upstream:   128/kbs
      Minimum Bitrates:
         Interleave Path:  downstream:     0/kbs,   upstream:     0/kbs
      Margin:              downstream:     3 db,    upstream:     3 db
      Interleave Delay:    downstream: 16000 usecs, upstream: 16000 usecs
      FEC Redundancy Bytes:
         Interleave Path:  downstream:    16,       upstream:    16
      R-S Codeword Size:   downstream:  auto,       upstream:  auto
      Trellis Coding:         Enabled
      Overhead Framing:       Mode 1
 Bit-Swap:               Disabled
Operating Mode:         Automatic
      Training Mode:          Standard

 SDSL profile parameters

 CAP profile parameters
```

Setting the Power Spectral Density Mask

This section describes how to set the ATU-C CAP and ATU-C FLEXI CAP power spectral density mask (PSDM) upstream and downstream values.

To set the ATU-C CAP and ATU-C FLEXI CAP power spectral density mask (PSDM) upstream and downstream values, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode, specifying the *profile-name* for which you want to set the PSDM value.
3.	DSLAM(cfg-dsl-profile)#**cap psdm downstream** *psdm* **upstream** *psdm*	Set the PSDM rate downstream and upstream for this profile.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Defaults

The default decibel values for PSDM rates are:

- –40 dB downstream
- –38 dB upstream

Examples

In this example, the command sets the CAP PSDM value at –37 dB downstream and –41 dB upstream for the profile “issis”.

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#cap psdm downstream -37 upstream -41
DSLAM(cfg-dsl-profile)#end
```

In this example, the command sets the CAP PSDM value to the default downstream and upstream settings of –40 dB and –38 dB for the profile “issis”.

```
DSLAM#configure terminal
DSLAM(config)#dsl-profile issis
DSLAM(cfg-dsl-profile)#no cap psdm downstream -40 upstream -38
DSLAM(cfg-dsl-profile)#end
```

Setting the ATU-C CAP CPE-Signature

You can set the customer premises equipment (CPE) signature for each configuration profile. To set the CAP CPE-signature for a desginated profile, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl-profile** *profile-name*	Go to the profile configuration mode and specify the *profile-name* for which you want to set the CPE signature value.

Step	Command	Task
3.	DSLAM(cfg-dsl-profile)#**cap cpe-signature** *0-255*	Set the CPE signature value.
4.	DSLAM(cfg-dsl-profile)#**end**	Exit from profile configuration mode.

Running the Chipset Self-Test

To run the DMT chipset self-test, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**dsl test atm** *slot/port* **self**	Run the self test on the specified slot and port.
3.	DSLAM#**show dsl interface atm** *slot/port*	Display the results of the self test.

This command runs a digital bit error-rate loopback test on the specified port. The run time for the self-test ranges from 3 seconds for the ATUC-1-4DMT card to 1 minute for the 4-port Flexi card.

To view the result of the self-test, use the command **show dsl interface atm** *slot/port*.

The output for this command includes the result of the last self-test, such as

```
Last Self-Test Result: NONE
```

The possible self-test results are PASSED, FAILED, RUNNING, and NONE. NONE means that a chipset self-test has not run since the port became operational. RUNNING means that the test is in progress.

Caution The chipset self-test disrupts port operation. If a port has trained or is training when this test begins, the port becomes untrained, the test executes, and the port retrains.

Example

This example runs the chipset self-test for port 1 on slot 6 and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#dsl test atm 6/1 self
DSLAM(config)#exit
DSLAM#show dsl interface atm 6/1
Port Status:
   Subscriber Name:          Circuit ID:
   IOS admin: UP     oper: UP     Card status: Present
   Last Change: 36352 days, 13 hrs, 51 min, 47 sec No. of changes: 0
   Line Status: TRAINED
   Test Mode:   NONE
Loopback: NONE

ADSL Chipset Self-Test: PASSED
CO Modem Firmware Version: 0x1319BE02

Configured:
```

```
DMT Profile Name: fast
Alarms Enabled: NO
```

Enabling and Disabling ATM Local Loopback

When you enable the loopback functionality, loopback cells are inserted on designated VPCs/VCCs. The NI-2 notifies you through the management information base (MIB) or Interim Local Management Interface (ILMI) if loopback cells do not return.

This section describes how to enable and disable ATM local loopback on a port.

To enable ATM local loopback on a port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Go to the interface configuration mode and specify the port for which you want to enable local loopback.
3.	DSLAM(config-if)#**loopback diagnostic**	Enable the loopback diagnostic for the selected port.
4.	DSLAM(config-if)#**end**	Exit from profile configuration mode.

To disable ATM local loopback on a port, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to the global configuration mode.
2.	DSLAM(config)#**interface atm** *slot/port*	Go to the interface configuration mode and specify the port for which you want to enable local loopback.
3.	DSLAM(config-if)#**no loopback diagnostic**	Disable the loopback diagnostic for the selected port.
4.	DSLAM(config-if)#**end**	Exit from profile configuration mode.

This command retrains the port if you change the parameter. Setting a parameter to its previous value does not cause a retrain. If a port is training when you change this parameter, the port untrains and retrains to the new parameter.

Example

This command disables ATM local loopback for port 1 on slot 0 and displays the results:

```
DSLAM#configure terminal
DSLAM(config)#interface atm 0/1
DSLAM(config-if)#no loopback diagnostic
DSLAM(config-if)#end
DSLAM#show dsl interface atm 0/1
Port Status:
   Subscriber Name:          Circuit ID:
   IOS admin: UP     oper: UP     Card status: Present
   Last Change: 36352 days, 13 hrs, 51 min, 47 sec No. of changes: 0
   Line Status: TRAINED
   Test Mode:   NONE
Loopback: NONE

ADSL Chipset Self-Test: NONE
CO Modem Firmware Version: 0x1319BE02
.
.
.
```

Displaying DSL and ATM Status

To display DSL and ATM status for a port, follow these steps:

Step	Command	Task
1.	DSLAM#**show dsl status** *slot/port*	Display the administrative and operational status of the port (up/down), the actual line rates, the subscriber name and circuit ID assigned to the port, and the subtend ID for the specified *slot/port*.
2.	DSLAM#**show dsl interface atm** *slot/port*	Display the information provided by **show dsl status**, plus configured profile parameters and actual parameter values for the specified *slot/port.*

Example

This example displays the DSL and ATM status for port 1 in slot 4:

```
DSLAM#show dsl status 4/1
DSLAM#show dsl interface atm 4/1
Port Status:
   Subscriber Name:          Circuit ID:
   IOS admin: UP     oper: DOWN     Card status: ATUC-1-4DMT
   Last Change: 00 days, 00 hrs, 12 min, 33 sec No. of changes: 684
   Line Status: NO CPE DETECTED
   Test Mode: NONE

ADSL Chipset Self-Test: NONE

CO Modem Firmware Version: 0x30CCBE05

Configured:
      DMT Profile Name: default
      Link Traps Enabled: NO
      Alarms Enabled: NO
```

```
      ATM Payload Scrambling: Enabled


 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:     4,        upstream:     6
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:    16,        upstream:     8
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick

Status:
    Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
    Attainable Aggregate
    Bitrates:
                            downstream:     0 kb/s,   upstream:     0 kb/s
    Margin:                 downstream:     0 dB,     upstream:     0 dB
    Attenuation:            downstream:     0 dB,     upstream:     0 dB
    Interleave Delay:       downstream:     0 usecs,  upstream:     0 usecs
    Check Bytes (FEC):
         Interleave Path:   downstream:     0,        upstream:     0
         Fast Path:         downstream:     0,        upstream:     0
    R-S Codeword Size:      downstream:     0,        upstream:     0
    Trellis Coding:           Not In Use
    Overhead Framing:         Mode 0
    Line Fault:               NONE
    Operating Mode:           Unknown
    Line Type:                Fast and Interleaved

    Alarms:
       status:                NONE

ATM Statistics:
   Interleaved-Path Counters:
      Cells:                downstream:          0     upstream:          0
      HEC errors:           downstream:          0     upstream:          0
      LOCD events:          near end:            0     far end:           0
   Fast-Path Counters:
      Cells:                downstream:          0     upstream:          0
      HEC errors:           downstream:          0     upstream:          0
      LOCD events:          near end:            0     far end:           0

DSL Statistics:
   Init Events:           341
   Transmitted Superframes: near end:            0     far end:           0
   Received Superframes:    near end:            0     far end:           0
   Corrected Superframes:   near end:            0     far end:           0
   Uncorrected Superframes: near end:            0     far end:           0

CPE Info
   Serial Number:          00000000
   Vendor ID:              0
```

```
Version Number:          0
```

Displaying Hardware Information

This section describes how to display information about the DSLAM hardware components.

To display a list of the cards in the chassis, the chassis type, and whether the power supply and fan interfaces are present, follow this step:

Step	Command	Task
1.	DSLAM#**show hardware**	Display the type of card in each slot in the chassis, the chassis type, and whether the power supply and fan interfaces are present.

To display the name of the card in the specified slot, follow this step:

Step	Command	Task
1.	DSLAM#**show hardware slot** *slot*	Display the name of the card in the specified slot.

To display the manufacturing information for the card in the slot: Chassis type, chassis name, manufacturer's name, H/W revision, Serial #, Asset ID, Alias, and CLEI code, follow this step:

Step	Command	Task
1.	DSLAM#**show hardware chassis**	Display the manufacturing information for the DSLAM: Chassis type, chassis name, manufacturer's name, H/W revision, Serial #, Asset ID, Alias, and CLEI code.

To display the online insertion and removal (OIR) status of the line cards, follow this step:

Step	Command	Task
1.	DSLAM#**show oir status** [*slot*]	Display the manufacturing information for the DSLAM: Chassis type, chassis name, manufacturer's name, H/W revision, Serial #, Asset ID, Alias, and CLEI code.

The **show oir status** command reports the status of line card slots in the DSLAM chassis. The reported status is one of the following:

- Loading: the line card in this slot is loading a new image, which typically takes about 2 minutes.
- Running: the line card in this slot is operating normally.
- Keepalive: the NI-2 is unable to communicate with the line card in this slot. The NI-2 keeps the line card in keepalive state for several seconds. If communication does not resume, the system assumes the card was removed.

When the NI-2 cannot communicate with a line card, the NI-2 provides no entry for the slot where the card is located. The **show oir status** command displays a history of attempts to communicate with the line card.

Examples

This example displays the physical card in the chassis and the chassis type and indicates if the power supply and fan interfaces are present:

```
DSLAM#show hardware

Chassis Type:C6160

Slot 1 :EMPTY                                   Slot 18:EMPTY
Slot 2 :EMPTY                                   Slot 19:ATUC-4FLEXICAP
Slot 3 :EMPTY                                   Slot 20:EMPTY
Slot 4 :EMPTY                                   Slot 21:ATUC-1-4DMT
Slot 5 :EMPTY                                   Slot 22:ATUC-4FLEXIDMT
Slot 6 :EMPTY                                   Slot 23:EMPTY
Slot 7 :EMPTY                                   Slot 24:EMPTY
Slot 8 :EMPTY                                   Slot 25:EMPTY
Slot 9 :EMPTY                                   Slot 26:EMPTY
Slot 10:NI-2-DS3-DS3                            Slot 27:EMPTY
Slot 11:EMPTY                                   Slot 28:EMPTY
Slot 12:STUC-4-2B1Q-DIR-1                       Slot 29:EMPTY
Slot 13:EMPTY                                   Slot 30:EMPTY
Slot 14:EMPTY                                   Slot 31:EMPTY
Slot 15:EMPTY                                   Slot 32:EMPTY
Slot 16:EMPTY                                   Slot 33:EMPTY
Slot 17:EMPTY                                   Slot 34:EMPTY

Fan Module 1: Present   2: Present

Power Supply Module 1: 6260-PEM-AC
Power Supply Module 2: 6260-PEM-AC
```

This example displays information on the cards in slots 20 and 21:

```
DSLAM#show hardware slot 20

Slot 20:EMPTY

DSLAM#show hardware slot 21

Slot 21: ATUC-1-4DMT

        Hardware Revision        : 1.0
        Part Number              : 800-05262-03
        Board Revision           : A0
        Deviation Number         : 0-0
        Fab Version              : 03
        PCB Serial Number        : SAL04300VR2
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        CLEI Code                : DML2GGCAAB
        Asset Identifier         :
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 40 01 53 41 01 00 C0 46 03 20 00 14 8E 03
          0x10: 42 41 30 80 00 00 00 00 02 03 C1 8B 53 41 4C 30
          0x20: 34 33 30 30 56 52 32 03 00 81 00 00 00 00 04 00
```

```
          0x30: C6 8A 44 4D 4C 32 47 47 43 41 41 42 CC 20 00 00
          0x40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          0x50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF
          0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

This example displays the manufacturing information for the DSLAM, including information on the NI2, backplane, I/O card, and power modules.

```
DSLAM#show hardware chassis

Chassis Type: C6260

NI2 Daughtercard EEPROM:
        Hardware Revision        : 1.0
        Part Number              : 73-3952-05
        Board Revision           : A0
        Deviation Number         : 0-0
        Fab Version              : 02
        PCB Serial Number        : 00010218817
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        Unknown Field (type 0086): 00 00 00 00
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 40 01 4F 41 01 00 82 49 0F 70 05 42 41 30
          0x10: 80 00 00 00 00 02 02 C1 8B 30 30 30 31 30 32 31
          0x20: 38 38 31 37 03 00 81 00 00 00 00 04 00 86 00 00
          0x30: 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x40: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x50: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

NI2 Motherboard EEPROM:
        Hardware Revision        : 1.0
        Part Number              : 800-05631-05
        Board Revision           : 01
        Deviation Number         : 0-0
        Fab Version              : 03
        PCB Serial Number        : 00010218817
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        CLEI Code                : unassigned
        Asset Identifier         : 00000000000000000000000000000000
        Processor type           : 00
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 40 01 94 41 01 00 C0 46 03 20 00 15 FF 05
          0x10: 42 30 31 80 00 00 00 00 02 03 C1 8B 30 30 30 31
          0x20: 30 32 31 38 38 31 37 03 00 81 00 00 00 00 04 00
          0x30: C6 8A 75 6E 61 73 73 69 67 6E 65 64 CC 20 30 30
          0x40: 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
          0x50: 30 30 30 30 30 30 30 30 30 30 30 30 30 09 00
          0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

BackPlane EEPROM:
        Hardware Revision        : 1.0
        Part Number              : 73-3999-05
        Board Revision           : A0
```

```
        Deviation Number         : 0-0
        Fab Version              : 04
        PCB Serial Number        : SAA04090051
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        Chassis Serial Number    : SCA041007X7
        CLEI Code                : DMM3BH0ERA
        Asset Identifier         :
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 41 01 00 82 49 0F 9F 05 42 41 30 80 00 00
          0x10: 00 00 02 04 C1 8B 53 41 41 30 34 30 39 30 30 35
          0x20: 31 03 00 81 00 00 00 00 04 00 C2 8B 53 43 41 30
          0x30: 34 31 30 30 37 58 37 C6 8A 44 4D 4D 33 42 48 30
          0x40: 45 52 41 CC 20 00 00 00 00 00 00 00 00 00 00 00
          0x50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          0x60: 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

I/O Card EEPROM:
        Hardware Revision        : 1.0
        Part Number              : 800-08690-01
        Board Revision           : 01
        Deviation Number         : 0-0
        Fab Version              : 01
        PCB Serial Number        : SAD04350CBB
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        Chassis MAC Address      : 0001.64ff.a97f
        MAC Address block size   : 1024
        CLEI Code                : ABCDEFGHIJ
        Asset Identifier         :
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 40 02 43 41 01 00 C0 46 03 20 00 21 F2 01
          0x10: 42 30 31 80 00 00 00 00 02 01 C1 8B 53 41 44 30
          0x20: 34 33 35 30 43 42 42 03 00 81 00 00 00 00 04 00
          0x30: C3 06 00 01 64 FF A9 7F 43 04 00 C6 8A 41 42 43
          0x40: 44 45 46 47 48 49 4A CC 20 00 00 00 00 00 00 00
          0x50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          0x60: 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Slot 1 Power Module EEPROM:
        Hardware Revision        : 1.0
        Part Number              : 34-1695-01
```

```
        Deviation Number         : 0-0
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        Chassis Serial Number    : 00000000562
        Power Supply Type        : AC
        CLEI Code                :
        Asset Identifier         :
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 41 01 00 82 22 06 9F 01 80 00 00 00 00 03
          0x10: 00 81 00 00 00 00 04 00 C2 8B 30 30 30 30 30 30
          0x20: 30 30 35 36 32 0B 00 C6 8A 00 00 00 00 00 00 00
          0x30: 00 00 00 CC 20 00 00 00 00 00 00 00 00 00 00 00
          0x40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          0x50: 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF
          0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Slot 2 Power Module EEPROM:
        Hardware Revision        : 1.0
        Part Number              : 34-1695-01
        Deviation Number         : 0-0
        RMA Test History         : 00
        RMA Number               : 0-0-0-0
        RMA History              : 00
        Chassis Serial Number    : 00000000552
        Power Supply Type        : AC
        CLEI Code                :
        Asset Identifier         :
        EEPROM format version 4
        EEPROM contents (hex):
          0x00: 04 FF 41 01 00 82 22 06 9F 01 80 00 00 00 00 03
          0x10: 00 81 00 00 00 00 04 00 C2 8B 30 30 30 30 30 30
          0x20: 30 30 35 35 32 0B 00 C6 8A 00 00 00 00 00 00 00
          0x30: 00 00 00 CC 20 00 00 00 00 00 00 00 00 00 00 00
          0x40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
          0x50: 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF
          0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0x90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
          0xF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Configuring ATM Interfaces

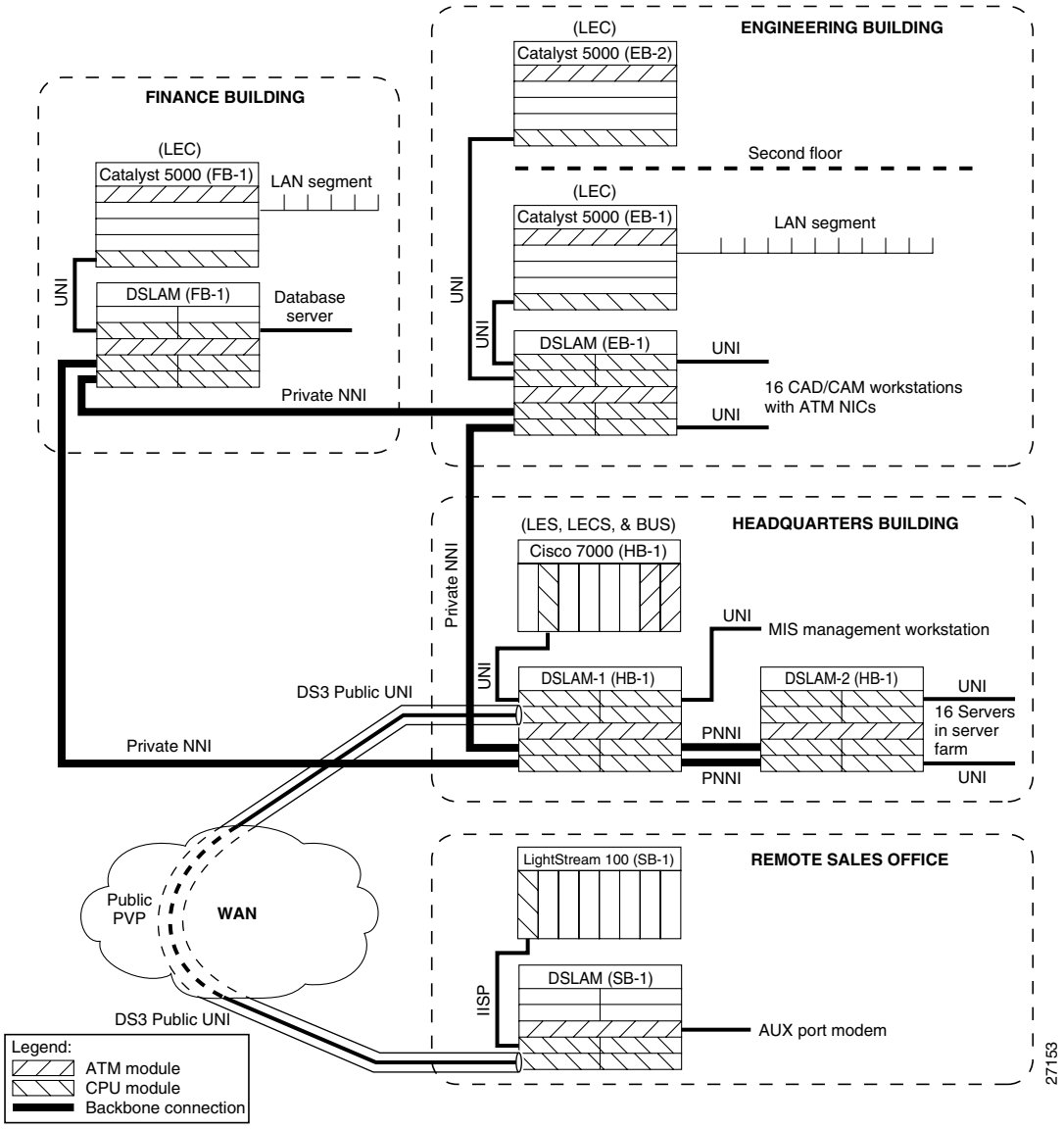
This chapter describes how to explicitly configure ATM network interface types. Explicitly configuring interfaces is the alternative to Interim Local Management Interface (ILMI) autoconfiguration, which senses the peer interface type and appropriately configures the Cisco 6000 family DSLAM interface.

Use the network configuration tasks described in this chapter to explicitly change your ATM DSLAM operation from the defaults, which are suitable for most networks. This chapter includes these sections:

Network Configuration Example

The sample network shown in Figure 8-1 illustrates some standard ATM interface configuration tasks you can perform after you complete the initial DSLAM configuration. See Chapter 3, "Initially Configuring the Cisco DSLAM."

Figure 8-1 Sample Network Configuration



The network configuration shown in Figure 8-1 is an example of a corporate campus ATM backbone network connecting three buildings with an ATM connection across the WAN to a remote sales office.

The remaining sections in this chapter describe a possible configuration of the network that appears in Figure 8-1.

Disabling Autoconfiguration

When an interface comes up initially, autoconfiguration determines the type of interface. To configure the interface protocol on an interface, you must first disable the autoconfiguration feature.

To disable autoconfiguration on an interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.vpt#*]	Select the interface to be configured.
2.	**no atm auto-configuration**	Disable autoconfiguration on the interface.

Example

In this example, autoconfiguration is disabled on interface ATM 0/1 and the results are as follows:

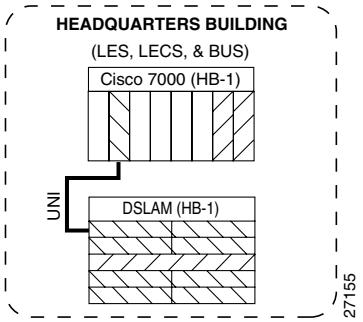
```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# no atm auto-configuration
DSLAM(config-if)#
%ATM-6-ILMINOAUTOCFG: ILMI(ATM0/1): Auto-configuration is disabled, current interface
parameters will be used at next interface restart.

DSLAM# atm interface atm 0/1

Interface:      ATM0/1        Port-type:    suni_dual
IF Status:      UP              Admin Status: up
Auto-config:    disabled        AutoCfgState: not applicable
IF-Side:        Network         IF-type:      NNI
Uni-type:       not applicable  Uni-version:  not applicable
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2b81.4000.0c80.8000.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      3        0       0       0        0       0           3                3
Logical ports(VP-tunnels):     0
Input cells:    234663          Output cells: 235483
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 153211, Output AAL5 pkts: 153626, AAL5 crc errors: 0
```

Configuring UNI Interfaces

The UNI specification defines communications between ATM end stations (such as workstations and routers) and ATM switches in private ATM networks. (The DSLAM functions as an ATM switch.)

To configure a UNI interface between the DSLAM (HB-1) in the headquarters building to the Cisco 7000 (HB-1) in the same building, use the **atm uni** command in interface configuration mode. Figure 8-2 shows a detail of this type of network connection.

Figure 8-2 Multiple Link UNI Example



To configure the UNI interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**no atm auto-configuration**	Disable autoconfiguration on the interface.
3.	**atm uni** [**side** {**network** | **user**}] [**type** {**private** | **public**}] [**version** {**3.0** | **3.1** | **4.0**}]	Configure the ATM UNI interface.

Note Each time you configure a change in the interface protocol (such as UNI, NNI, or IISP), side, or version, ATM signaling and ILMI is restarted on the interface. When you restart ATM signaling, the DSLAM clears all switched virtual connections (SVCs) across the interface. Permanent virtual connections are not affected.

Example

This example disables autoconfiguration on ATM interface 0/1 and to configure the interface as the user side of a private UNI running Version 4.0:

```
DSLAM(HB-1)(config)# interface atm 0/1
DSLAM(HB-1)(config-if)# no atm auto-configuration
DSLAM(HB-1)(config-if)#
%ATM-6-ILMINOAUTOCFG: ILMI(ATM0/1): Auto-configuration is disabled, current interface
parameters will be used at next interface restart.
DSLAM(HB-1)(config-if)# atm uni side user type private version 4.0
DSLAM(HB-1)(config-if)#
%ATM-5-ATMSOFTSTART: Restarting ATM signalling and ILMI on ATM0/1.
```

To show the ATM interface UNI configuration, use this EXEC command:

Command	Task
show atm interface atm *slot/port*	Show the ATM switch configuration.

Example

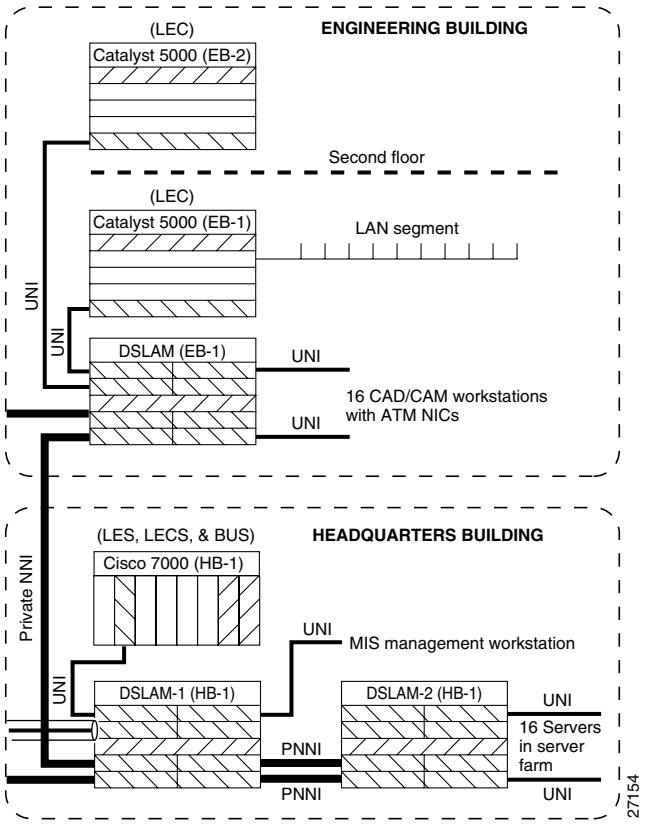
This example displays the ATM interface 0/1 UNI configuration:

```
DSLAM(HB-1)# show atm interface atm 0/1

Interface:      ATM0/1        Port-type:    suni_dual
IF Status:      UP              Admin Status: up
Auto-config:    disabled        AutoCfgState: not applicable
IF-Side:        User            IF-type:      UNI
Uni-type:       Private         Uni-version:  V4.0
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2b81.4000.0c80.8000.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      2        0       0       0        0       0           2                2
Logical ports(VP-tunnels):     0
Input cells:    234810          Output cells: 235618
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 153296, Output AAL5 pkts: 153712, AAL5 crc errors: 0
```

Configuring NNI Interfaces

This section describes the configuring of a Network-to-Network Interface (NNI) connection between two switches. The example in this section involves the configuring of a Private NNI (PNNI) interface from the DSLAM (HB-1) in the headquarters building to the DSLAM (EB-1) in the Engineering building, as shown in Figure 8-3.

Figure 8-3 Private NNI Interface Example



You must configure PNNI connections between the ATM switches to allow for route discovery and topology analysis between the switches. To configure the NNI interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**no atm auto-configuration**	Disable autoconfiguration on the interface.
3.	**atm nni**	Configure the ATM NNI interface.

Note Each time you configure a change in the interface protocol (such as UNI, NNI, or IISP), side, or version, ATM signaling and ILMI is restarted on the interface. When you restart ATM signaling, the DSLAM clears all switched virtual connections (SVCs) across the interface. Permanent virtual connections are not affected.

Example

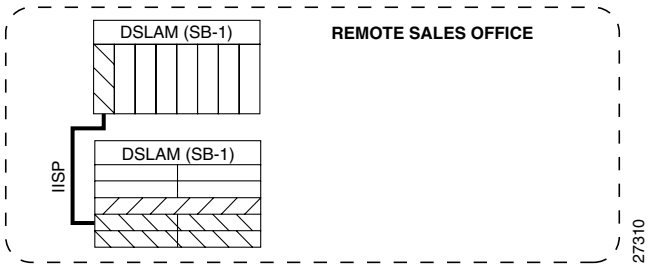
This example configures ATM interface 0/1 on the DSLAM located in the headquarters building as an NNI interface and displays the results:

```
DSLAM(HB-1)(config)# interface atm 0/1
DSLAM(HB-1)(config-if)# no atm auto-configuration
DSLAM(HB-1)(config-if)#
%ATM-6-ILMINOAUTOCFG: ILMI(ATM0/1): Auto-configuration is disabled, current interface
parameters will be used at next interface restart.
DSLAM(HB-1)(config-if)# atm nni
DSLAM(HB-1)(config-if)#
%ATM-5-ATMSOFTSTART: Restarting ATM signalling and ILMI on ATM0/1.

DSLAM(HB-1)# show atm interface atm 0/1

Interface:      ATM0/1          Port-type:    suni_dual
IF Status:      UP              Admin Status: up
Auto-config:    disabled        AutoCfgState: not applicable
IF-Side:        Network         IF-type:      NNI
Uni-type:       not applicable  Uni-version:  not applicable
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2b81.4000.0c80.8000.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      3        0       0       0        0       0           3                3
Logical ports(VP-tunnels):     0
Input cells:    234911          Output cells: 235695
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 153346, Output AAL5 pkts: 153764, AAL5 crc errors: 0
```

Configuring IISP Interfaces

This section describes how to configure the Interim Interswitch Signaling Protocol (IISP) interface from the DSLAM (SB-1) in the Remote Sales building to the DSLAM (SB-1) in the same building. Figure 8-4 shows an example of this type of network configuration.

Figure 8-4 IISP Network Segment Example



Some ATM switches do not support the Private Network-to-Network Interface (PNNI) protocol. Switched virtual circuit (SVC) support can be provided by configuring the interface to use IISP.

To configure the IISP interfaces in Figure 8-4, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.sub_inter#*]	Select the interface to be configured.
2.	**no atm auto-configuration**	Disable autoconfiguration on the interface.
3.	**atm iisp** [**side** {**network** | **user**}] [**version** {**3.0** | **3.1** | **4.0**}]	Configure the ATM IISP interface.
4.	**exit**	Exit interface configuration mode.
5.	**atm route prefix** *atm-address-prefix* **atm** *slot/port*[*.sub_inter#*]	Configure the ATM route address prefix.

Note Each time you configure a change in the interface protocol (such as UNI, NNI, or IISP), side, or version, ATM signaling and ILMI is restarted on the interface. When you restart ATM signaling, the DSLAM clears all switched virtual connections (SVCs) across the interface. Permanent virtual connections are not affected.

Example

This example configures ATM interface 0/1 on the DSLAM (SB-1) located in the Remote Sales building with these parameters, and displays the results:

- No autoconfiguration
- IISP
- Side as user
- ATM route address prefix as 47.0091.8100.0000.0000.0ca7.ce01

```
DSLAM(SB-1)(config)# interface atm 0/1
DSLAM(SB-1)(config-if)# no atm auto-configuration
DSLAM(SB-1)(config-if)#
%ATM-6-ILMINOAUTOCFG: ILMI(ATM0/1): Auto-configuration is disabled, current interface
parameters will be used at next interface restart.
DSLAM(SB-1)(config-if)# atm iisp side user
DSLAM(SB-1)(config-if)#
%ATM-5-ATMSOFTSTART: Restarting ATM signalling and ILMI on ATM0/1.
DSLAM(SB-1)(config-if)# exit
DSLAM(SB-1)(config)# atm route 47.0091.8100.0000.0000.0ca7.ce01 atm 0/1

DSLAM(SB-1)(config)# show atm int 0/1

Interface:      ATM0/1        Port-type:    suni_dual
IF Status:      DOWN            Admin Status: down
Auto-config:    disabled        AutoCfgState: not applicable
IF-Side:        User            IF-type:      IISP
Uni-type:       not applicable  Uni-version:  V3.0
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2b81.4000.0c81.8000.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      2        0       0       0        0       0           2                0
Logical ports(VP-tunnels):     0
Input cells:    0               Output cells: 0
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 0, Output AAL5 pkts: 0, AAL5 crc errors: 0
```

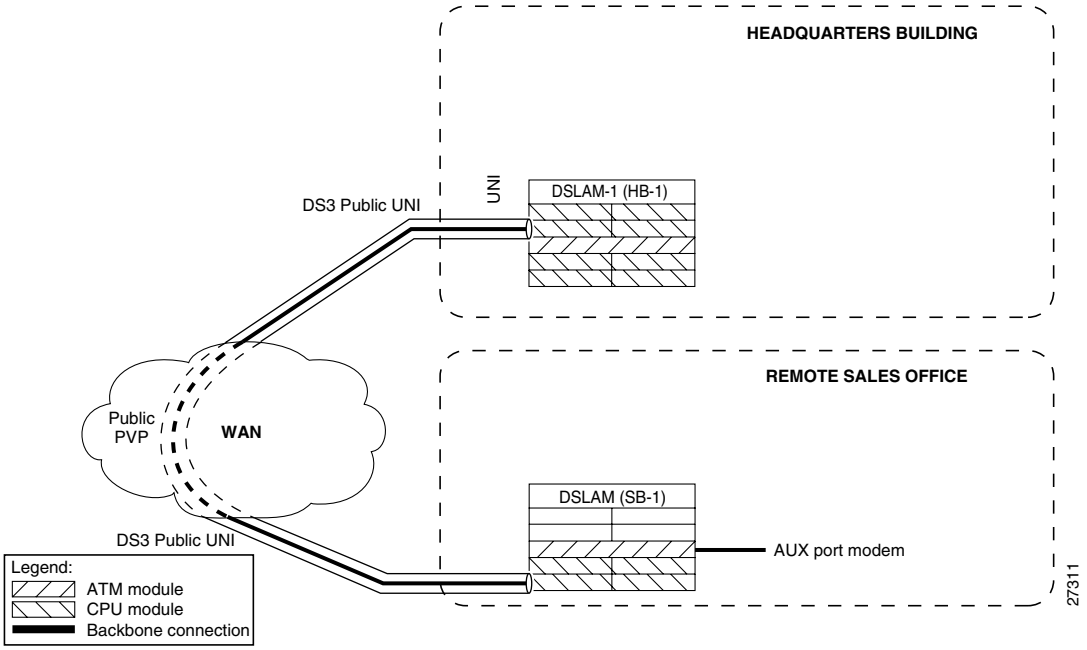
Configuring a Public Network Tunnel Interface

This section describes how to configure the DS3 public UNI ATM connection as a virtual path (VP) tunnel from the headquarters building across the WAN to the Remote Sales building.

Note A VP tunnel is configured as a VP of a particular service category. Only virtual circuits (VCs) of that service category can transit the tunnel.

Figure 8-5 shows a detail of the sample network you are configuring.

Figure 8-5 Public VP Tunnel Network Example



Public DS3 carriers can interconnect switches using permanent VPs across their networks. To support signaling across the public network between the DSLAM (HB-1) in the headquarters building and the Remote Sales building the DSLAM (SB-1), you must configure a VP tunnel.

Assigning VPI Values to Shaped VP Tunnels

If you configure VP tunnels with traffic shaping, you can use only 32 VPIs, even though the full range of VPI values is 0 to 255. If you have not yet assigned any VPIs, all values from 0 to 255 are available. Once you start assigning VPIs, however, the assigned VPIs limit the VPIs that remain. (You assign VPIs using the **atm pvp** or **atm pvc** commands.)

After a particular VPI value is assigned to a shaped VP tunnel, every 32nd VPI value above and below the first one is eliminated—that is, the original value modulo 32. For example, if you assign VPI 94 to a shaped VP tunnel, the following VPI values become unavailable for any purpose: 30, 62, 126, 158, 190, and 222.

To avoid problems, choose a block of 32 consecutive VPI values (for example, 0 to 31 or 101 to 132). The software rejects invalid VPI values.

To configure a VP tunnel connection, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**atm connection-traffic-table-row** [**index** *row-index*] **cbr pcr** *rate* [**tolerance** *cell-count*]	Optionally, configure the connection-traffic-table-row index for any nondefault index values.
2.	**interface atm** *slot/port*	Select the physical interface to be configured.
3.	**atm pvp** *vpi* [**rx-cttr** *index*] [**tx-cttr** *index*]	Configure an interface permanent virtual path (PVP) leg.

Step	Command	Task
4.	**exit**	Change to configuration mode.
5.	**interface atm** *slot/port.vpt#*	Create a VP tunnel using a VPT number that matches the PVP leg virtual path identifier (VPI).

Note You must configure the row index for no-default **rx-cttr** and **tx-cttr** before you use this optional parameter.

Examples

This example configures the ATM VP tunnel on the DSLAM (HB-1) located in the headquarters building at interface 0/1, VPI 99 to the DSLAM (SB-1) located in the Remote Sales building at interface 0/1, VPI 99:

```
DSLAM(HB-1)(config)# interface atm 0/1
DSLAM(HB-1)(config-if)# atm pvp 99
DSLAM(HB-1)(config-if)# exit
DSLAM(HB-1)(config)# interface atm 0/1.99
DSLAM(HB-1)(config-subif)# end
DSLAM(HB-1)#
%SYS-5-CONFIG_I: Configured from console by console
```

This example configures the ATM VP tunnel on the DSLAM (SB-1), located in the Remote Sales building at interface 0/1, VPI 99:

```
DSLAM(SB-1)(config)# interface atm 0/1
DSLAM(SB-1)(config-if)# atm pvp 99
DSLAM(SB-1)(config-if)# exit
DSLAM(SB-1)(config)# interface atm 0/1.99
DSLAM(SB-1)(config-subif)# end
DSLAM(SB-1)#
%SYS-5-CONFIG_I: Configured from console by console
```

To show the ATM virtual interface configuration, use this EXEC command:

Command	Task
show atm interface [**atm** *slot/port*[*.vpt#*]]	Show the ATM interface configuration.

Example

This example displays the configuration of the DSLAM (HB-1), located in the headquarters building at interface 0/1:

```
DSLAM(HB-1)# show atm interface atm 0/1

Interface:      ATM0/1          Port-type:    vp tunnel
IF Status:      UP              Admin Status: up
Auto-config:    enabled         AutoCfgState: waiting for response from peer
IF-Side:        Network         IF-type:      UNI
Uni-type:       Private         Uni-version:  V3.0
Max-VPI-bits:   0               Max-VCI-bits: 14
Max-VP:         0               Max-VC:       16383
```

```
Signalling:     Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2b81.4000.0c80.8000.63
Configured virtual links:
  PVCLs SoftVCLs   SVCLs  Total-Cfgd  Installed-Conns
      4        0       0           4                4
```

Configuring Signaling VPCI for PVP Tunnels

To specify the value of the virtual path connection identifier (VPCI) that is to be carried in the signaling messages within a VP tunnel, use the **atm signalling vpci** interface configuration command.

Note By default, the VPCI is the same as the VPI on the ATM switch.

The connection identifier information element (IE) is used in signaling messages to identify the corresponding user information flow. The connection identifier IE contains the VPCI and VCI.

For example, if you want to configure a PVP tunnel connection from a DSLAM on VPI 2, VCI X, to a router with a virtual path switch in between, the signaling message must contain connection ID, VPI 2, VCI X. Because the PVP tunnel at the router end is on VPI 3, VCI X, the connection is refused. By configuring VPCI to 3, you can configure the signaling message explicitly to contain connection ID VPI 3, VCI X, instead of containing VPI 2, VCI X.

You can also use this command to support virtual User-Network Interface (UNI) connections.

To configure a VP tunnel connection signaling VPCI, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port.vpt#*	Select the subinterface.
2.	**atm signalling vpci** *vpci_number*	Configure the atm signaling VPCI number. The range is from 0 to 255.

Example

This example configures a PVP tunnel on ATM interface 0/1, PVP 99, and then configures the connection ID VCPI as 0 in interface configuration mode.

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm pvp 99
Switch(config-if)# exit
Switch(config)# interface atm 0/1.99
Switch(config-subif)# atm signalling vpci 0
```

To confirm the PVP tunnel VPCI configuration, use this privileged EXEC command:

Command	Task
show running-config	Show the PVP tunnel interface configuration.

Deleting VP Tunnels

To delete a VP tunnel connection, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**no interface atm** *slot/port*[*.vpt#*]	Delete the interface.
2.	**interface atm** *slot/port*	Select the physical interface to be modified.
3.	**no atm pvp** *vpi*	Delete the interface PVP half-leg.

Example

This example deletes subinterface 99 at ATM interface 0/1 and the PVP half-leg 99 on the DSLAM (HB-1) and displays the results:

```
DSLAM(HB-1)(config)# no interface atm 0/1.99
DSLAM(HB-1)(config)# interface atm 0/1
DSLAM(HB-1)(config-if)# no atm pvp 99

DSLAM(HB-1)# show interface atm 0/1

Interface:      ATM0/1        Port-type:    suni_dual
IF Status:      UP              Admin Status: up
Auto-config:    enabled         AutoCfgState: completed
IF-Side:        Network         IF-type:      NNI
Uni-type:       not applicable  Uni-version:  not applicable
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2b81.4000.0c80.8000.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      3        0       0       0        0       0           3                3
Logical ports(VP-tunnels):     0
Input cells:    233651          Output cells: 234465
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 152555, Output AAL5 pkts: 152967, AAL5 crc errors: 0
```

Configuring a PVC to a VP Tunnel

To configure an endpoint of a permanent virtual circuit (PVC) to a previously created PVP tunnel, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the physical interface to be configured.
2.	**atm pvc** *vpi vci* [**upc** *upc*] [**pd** *pd*] [**rx-cttr** *index*] [**tx-cttr** *index*] **interface atm** *slot/port*[*.vpt#*] *vpi vci* [**upc** *upc*]	Configure the PVC with the VPI of the tunnel leg matching the tunnel VPT number.

These restrictions apply to an endpoint of a PVC-to-PVP tunnel subinterface:

- The VPI number of the tunnel leg of any PVC connection must match the VPT number of the tunnel.
- The service class (for example, constant bit rate [CBR]), variable bit rate [VBR], unspecified bit rate [UBR], as specified by the connection-traffic-table-row [CTTR]) of any PVC connections must match the service class for the rows selected for the tunnel PVP.
- For service classes other than UBR and available bit rate (ABR), the peak cell rates of all PVCs must be within the peak cell rate of the tunnel PVP. This setup requires new CTTR rows to be defined for CBR or VBR PVCs, with peak cell rates that are less than the intended tunnel PVP.

Example

This example shows you how to configure the example tunnel ATM0/1.99 with a UBR PVC from interface ATM 0/1 to the tunnel at ATM interface 0/1.99, and displays the results:

```
DSLAM(HB-1)(config)# interface atm 0/1
DSLAM(HB-1)(config-if)# atm pvc 0 50 interface atm 0/1.99 99 40

DSLAM(HB-1)# show atm vc interface atm 0/1
Interface    VPI   VCI   Type    X-Interface  X-VPI X-VCI  Encap Status
0/1     0     5      PVC     0/1     0     41    QSAAL  UP
0/1     0     16     PVC     0/1     0     33    ILMI   UP
0/1     0     50     PVC     ATM0/1.99  99    40           UP
```

Configuring a VPI or VCI Range for SVPs or SVCs

You can configure a virtual path identifier or virtual channel identifier (VPI or VCI) range for switched virtual circuits or switched virtual paths (SVCs or SVPs). This feature allows you to

- Specify ranges for SVCs and SVPs.
- Avoid VPI or VCI conflicts when you attempt to set up soft PVCs or soft PVPs. For example, if you specify a soft PVC with VPI 0 and VCI 50 on the destination interface, an SVC on that interface might have already taken VPI 0 and VPI 50 just before the soft PVC setup message arrives at the destination interface. In this case, the soft PVC is rejected because VPI 0 and VPI 50 are already taken. By specifying the VPI or VCI range for SVPs or SVCs, you can avoid connection setup rejections.

You can still configure PVCs and PVPs in any supported range, including any VPI or VCI range you configured for SVCs or SVPs.

Note ILMI Version 4.0 supports this feature.

The default maximum switched virtual path connection (SVPC) VPI is equal to the maximum VPI supported on the interface. You can change the maximum SVPC VPI by entering the **atm svpc vpi max** *value* command. Substitute *value* with:

- A number in range of 0 to 3 for 25-Mbps interfaces.
- A number in range of 0 to 255 for all other interfaces except logical interfaces, which have a fixed value of 0.

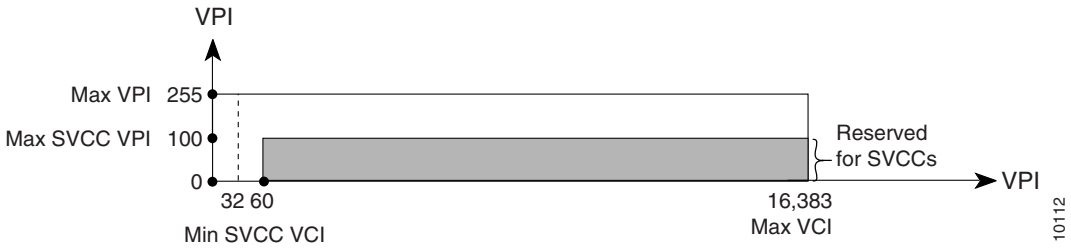
The default maximum switched virtual channel connection (SVCC) VPI is equal to the maximum VPI supported on the interface. You can change the maximum SVCC VPI by entering the **atm svcc vpi max** *value* command. Substitute *value* with:

- A number in the range of 0 to 3 for 25-Mbps interfaces.
- A number in the range of 0 to 255 for all other interfaces except logical interfaces, which have a fixed value of 0.

The default minimum SVCC VCI is equal to 35. You can change the minimum SVCC VCI by entering the **atm svcc vci min** *value* command. Substitute *value* with a number in the range of 32 to 4095.

In the example shown in Figure 8-6, the maximum SVCC VPI is 100 and the minimum SVCC VCI is 60. Therefore, VPIs 0 through 100 and VCIs 60 through 16,383 are reserved for SVCCs.

Figure 8-6 Sample SVCC VPI or VCI Range



Each interface negotiates the local values for the maximum SVPC VPI, maximum SVCC VPI, and minimum SVCC VCI with the peer's local value during ILMI initialization. The negotiated values determine the ranges for SVPs and SVCs. If the peer interface does not support these objects or autoconfiguration is turned off on the local interface, the local values determine the range.

To configure a VPI or VCI range for SVCs or SVPs, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the physical interface to be configured.
2.	**atm svpc vpi max** *value*	Configure the maximum VPI value for a SVPC.
3.	**atm svcc vpi max** *value*	Configure the maximum VPI value for a SVCC.
4.	**atm svcc vci min** *value*	Configure the minimum VCI value for a SVCC.

To confirm the VPI or VCI range configuration, use one of these commands:

Command	Task
show atm interface atm *slot/port*	Show the ATM interface configuration.
show atm ilmi-status atm *slot/port*	Show the ILMI status on the ATM interface.

Examples

This example confirms the VPI or VCI range configuration on an ATM interface. The values displayed for ConfMaxSvpcVpi, ConfMaxSvccVpi, and ConfMinSvccVci are local values. The values displayed for CurrMaxSvpcVpi, CurrMaxSvccVpi, and CurrMinSvccVci are negotiated values.

```
Switch# show atm interface atm 0/0

Interface:      ATM0/0        Port-type:      suni_dual
IF Status:      DOWN            Admin Status:   down
Auto-config:    enabled         AutoCfgState:   waiting for response from peer
IF-Side:        Network         IF-type:        UNI
Uni-type:       Private         Uni-version:    V3.0
Max-VPI-bits:   8               Max-VCI-bits:   14
Max-VP:         255             Max-VC:         16383
ConfMaxSvpcVpi: 100             CurrMaxSvpcVpi: 100
ConfMaxSvccVpi: 100             CurrMaxSvccVpi: 100
ConfMinSvccVci: 60              CurrMinSvccVci: 60
Svc Upc Intent: pass            Signalling:     Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0040.0b0a.2a81.4000.0c80.0000.00
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   TVCLs   PVPLs SoftVPLs   SVPLs Total-Cfgd Inst-Conns
      3        0       0       0       0        0       0           3          0
Logical ports(VP-tunnels):     0
Input cells:    0               Output cells:   0
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 0, Output AAL5 pkts: 0, AAL5 crc errors: 0
```

This example confirms the peer's local values for VPI or VCI range configuration by displaying the ILMI status on an ATM interface:

```
Switch# show atm ilmi-status atm 0/0

Interface : ATM0/0 Interface Type : Private NNI
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
Addr Reg State:   UpAndNormal
Peer IP Addr:     172.20.40.232   Peer IF Name:     ATM0/0
Peer MaxVPIbits:  8               Peer MaxVCIbits:  14
Peer MaxVPCs:     255             Peer MaxVCCs:     16383
Peer MaxSvccVpi:  255             Peer MinSvccVci:  255
Peer MaxSvpcVpi:  48
Configured Prefix(s) :
47.0091.8100.0000.0010.11ba.9901
```

Note Note that the **show atm ilmi-status** command displays the information above only if the peer supports it.

Configuring Resource Management

This chapter describes resource management functions for Cisco DSLAMs with NI-2 cards. Resource management involves modeling and managing switch, interface, and connection resources. Such resources include equivalent bandwidth and buffering to support the provision of specified traffic classes.

This chapter contains these sections:

Resource Management Functions

The DSLAM resource management software provides these functions:

- Network management interface—Includes operational configuration changes (which take place immediately), proposed configuration changes (which take place on restart), user interface, and status.

- Default quality of service (QoS) objective table management—Because User-Network Interface 3 (UNI 3) signaling does not provide information elements to signal QoS values, resource management provides a table that contains default values for QoS.
- Connection Traffic Table (CTT) management—Rather than store traffic parameters for each connection in that connection's data structure, resource management manages a table of connection traffic parameters, used by network and connection management.
- Resource Call Admission Control (RCAC)—Determines whether a virtual channel connection/virtual path connection (VCC/VPC) can be admitted (allowed to be set up), based on the available connection resources and requested traffic characteristics.
- Logical interface creation and deletion.
- Private Network-Network Interface (PNNI) metrics—Resource management supplies PNNI with link metrics for connection routing.

Creating a Connection Traffic Table Row for PVC Traffic Parameters

To properly manage your connection resources, you must create a row in the Connection Traffic Table (CTT) for each unique combination of traffic parameters used on a PVC flow.

To create a row in the CTT for CBR and UBR traffic parameters, use these commands:

Step	Command	Task
1.	**atm connection-traffic-table-row** [**index** *row-index*] **cbr pcr** *pcr-value* [**cdvt** *cdvt-value*]	Selects the row-index, pcr-value, and cdvt-value.
2.	**atm connection-traffic-table-row** [**index** *row-index*] **ubr pcr** *pcr-value* [**mcr** *mcr-value*] [**cdvt** *cdvt-value*]	Selects the row-index, pcr-value, mcr-value, and cdvt-value.
3.	**atm connection-traffic-table-row** [**index** *row-index*] **pcr** *pcr-value* [**vbr-rt** | **vbr-nrt**] [**scr10 scr0** *scr-value*] [**mbs** *mbs-value*] [**cdvt** *cdvt-value*]	Selects the row-index, pcr-value, scr-value, mbs-value, and cdvt-value.

Note The DSLAM does not distinguish between cells that have SCR = 10 and cells that have SCR = 0 with respect to the policing of incoming cells. Therefore there is effectively no difference in the configuration when SCR10 or SCR0 is chosen. This also applies to any values that display as the result of a **show** command.

Note No traffic shaping or policing is available in the downstream direction.

Example

This example creates the index row in the CTT for UBR traffic parameters with a row index of 15 and a pcr value of 424, and displays the results:

```
DSLAM# atm connection-traffic-table-row index 15 ubr pcr 424
DSLAM# show atm connection-traffic-table
```

```
Row       Service-category    pcr         scr/mcr       mbs          cdvt
1              ubr           7113539        none                      none
2              cbr              424                                   none
3              vbr-rt           424          424          50          none
4              vbr-nrt          424          424          50          none
15             ubr              424         none                       none
64000          cbr             1741                                   none
2147483645*    ubr                0         none                       none
2147483646*    ubr                1         none                       none
2147483647*    ubr           7113539        none                       none
```

Enabling and Disabling the clp-drop Flag

This section describes how to enable or disable the clp-drop flag for selected traffic parameters.

To enable or disable the clp-drop flag, follow these steps:

Step	Command	Task
1.	DSLAM# **configure terminal**	Go to the global command level.
2.	DSLAM(config)# **atm clp-drop** {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr**} {**off** | **on**}	Enable or disable the clp-drop flag for selected traffic parameters.

When the clp-drop flag is enabled, the software drops cells when the specified service-category queues reach 50 percent of the discard threshold limit.

Example

This example enables the clp-drop flag for ubr traffic and displays the results:

```
DSLAM# configure terminal
DSLAM(config)# atm clp-drop ubr on
DSLAM(config)# exit
DSLAM# show atm resource
Resource configuration:
    Over-subscription-factor 8  Sustained-cell-rate-margin-factor 64%
    Abr-mode:   relative-rate

Subtended Port Input queue Configuration:
   Max sizes: 5000  cbr, 1024  vbr-rt, 8192  vbr-nrt, 8192  ubr
   Discard threshold: 75% cbr, 50% vbr-rt, 50% vbr-nrt, 50% ubr

>CLP Drop configuration:
  cbr : off,  vbr-rt : off,  vbr-nrt : off,  ubr : on
```

Queueing and Buffering

The DSLAM architecture has two stages of queueing and buffering that guarantee fairness across subtended interfaces:

- Upstream cells from the local modem shelves, as well as upstream cells from the subtended interfaces, are first queued in the input queues. There are separate sets of input queues for each of the 13 possible subtended shelves. The upstream cells are enqueued based on the shelf number they originated from.
- The input queues are then emptied to output queues in a manner that ensures fairness and the QoS guarantees. All downstream cells and all cells to and from the local management processor bypass the input queues and go directly to the output queues.

The sections that follow describe how to manage the sizes of these input queues.

Note Queue sizes are based on the number of cells, but the number of cells must be a power of 2.

Configuring the Input Queue Discard Threshold

This section describes how to set the input queue discard threshold value for subtended ports for selected traffic parameters.

Note Cisco recommends that you leave the input queue discard thresholds set to their default values, which are adequate for most configurations.

The **atm input-threshold** command controls the discard threshold settings for up to 52 input queues—one queue for each of four traffic types on each of up to 13 nodes in a subtending group:

(13 shelves)*(4 priorities) = 52 input queues

You can figure each priority differently, but all 13 of the potential shelves share the same queue size at any particular priority. Both the point at which all further cells are dropped, and the point at which EPD drops begin are based on the EPD setting.

The behavior of the input queues is affected not only by the input queue discard threshold settings, but also by the setting of the intelligent packet discard (PD) feature, which is controlled with the **atm pvc** command. The PD setting determines whether the system performs packet-based discards or cell-based discards:

- When PD is enabled, the system performs packet-based discards—that is, when discarding is triggered, the system drops data from the first cell dropped, up to the end of the current AAL5 packet.

  This discard method includes policer and partial packet discard (PPD) drops, or entire AAL5 packets (for early packet discard (EPD) drops). The system accepts or rejects subsequent data on a packet-by-packet basis.
- When PD is disabled, the system performs cell-based discards—that is, when discarding, the system drops a cell at a time, and accepts or rejects subsequent data on a cell-by-cell basis. Cell-based discarding is the default behavior.

The PD setting applies to all discards, whether for reasons of queue exhaustion or policing. PD is disabled by default; use the command **atm pvc** *vpi vci* **pd** {**on** | **off**} to enable or disable it.

The input queue discard thresholds work as follows:

- If packet-based discard is in force (the PD feature is enabled), the input queue absorbs packets until the queue reaches the **epd** threshold. At that point, the queue absorbs the remainder of the current packet, as long as doing so does not cause the queue to fill completely. (The total queue size equals **epd** value plus **drop** value.)

After it reaches the **epd** threshold, the queue drops all subsequent packets until the queue's contents drop below the **epd** threshold. If the queue fills completely before the current packet finishes, then PPD occurs.

- If cell-based discard is in force (the PD feature is disabled), add the **epd** and **drop** threshold values to determine the input queue size. When the queue is full, it drops all subsequent cells until its contents fall below the combined threshold value.

If packet-based discard is in force, you can implicitly configure the input queue discard thresholds for either EPD or PPD. For EPD, configure a **drop** threshold value that is large enough to allow most packets to enter the queue. Appropriate values for this purpose vary by traffic type, but see the thresholds in the table of defaults, below, for examples of EPD settings. For PPD, configure a small **drop** threshold value. This forces the system to discard the remainder of the packet that fills up the queue.

To set input queue sizes, use the **atm input-queue** command, as described in the "Configuring Modem Port Input Maximum Queue Size" section on page 9-158.

The default input queue discard threshold varies by interface type and by traffic priority, as shown here:

Interface	Queue Segment	cbr	vbr-rt	vbr-nrt	ubr
DS3	**epd**	512 cells	512 cells	4096 cells	4096 cells
	drop	512 cells	512 cells	4096 cells	4096 cells
	Total queue	1024 cells	1024 cells	8192 cells	8192 cells
OC-3c	**epd**	2048 cells	2048 cells	8192 cells	8192 cells
	drop	2048 cells	2048 cells	8192 cells	8192 cells
	Total queue	4096 cells	4096 cells	16384 cells	16384 cells

To set the input queue EPD threshold value for subtended ports for selected traffic parameters, follow these steps:

Step	Command	Task
1.	`DSLAM#` **configure terminal**	Go to the global command level.
2.	`DSLAM(config)#` **atm input-threshold** {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr**} {**epd** | **drop**} *t-value*	Set the input queue EPD threshold value (*t-value*) for the selected traffic parameters.

Example

This example shows you how to set the input queue EPD threshold CBR traffic for subtended ports to 32Kb and displays the results:

```
DSLAM# configure terminal
DSLAM(config)# atm input-threshold cbr epd 32000

DSLAM# show atm interface resource a0/2 !(Subtended Port output)
Resource Management configuration:
    Output queues:
        Max sizes(derived): 4096  cbr, 4096  vbr-rt, 32768  vbr-nrt, 32768  ubr
        EPD  threshold: 2048 cbr, 2048 vbr-rt, 16384 vbr-nrt, 16384 ubr
```

```
        Drop threshold: 2048 cbr, 2048 vbr-rt, 16384 vbr-nrt, 16384 ubr
    Subtended Input queues:
        Max queue sizes(Derived): 1024  cbr, 1024  vbr-rt, 8192  vbr-nrt, 8192  ubr
        EPD  threshold: 32000 cbr, 512 vbr-rt, 4096 vbr-nrt, 4096 ubr
        Drop threshold: 512 cbr, 512 vbr-rt, 4096 vbr-nrt, 4096 ubr
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 0 kilometers
<output deleted>
```

Configuring the Interface Queue Thresholds

The **atm output-threshold** command controls the discard threshold settings for up to 1040 output queues. You can specify the output queue discard thresholds for the different levels of service and configure them on each interface queue. You can configure these queue thresholds:

- Output queue cell loss priority (CLP)
- Packet discard (PD) threshold

The command **atm output-threshold** works on the output queues. Unlike the input queues, each output port has a unique setting specific to one port. As with the input queue settings, each priority can be set independently, and you can set both an EPD threshold and a maximum queue size.

These queue thresholds can be changed at any time. The result changes the threshold for all connections of that service category using the interface for output and for any subsequent connections.

Note Cisco recommends that you leave the output queue discard thresholds set to their default values, which are adequate for most configurations.

The behavior of the output queue is controlled not only by the output queue discard threshold settings, but also by the setting of the intelligent packet discard (PD) feature, which is controlled with the **atm pvc** command.

The PD setting determines whether the system performs packet-based discards or cell-based discards:

- When PD is enabled, the system performs packet-based discards—that is, when discarding is triggered, the system drops data from the first cell dropped, up to the end of the current AAL5 packet.

  This discard method includes policer and partial packet discard (PPD) drops, or entire AAL5 packets (for early packet discard (EPD) drops). The system accepts or rejects subsequent data on a packet-by-packet basis.

- When PD is disabled, the system performs cell-based discards—that is, when discarding, the system drops a cell at a time, and accepts or rejects subsequent data on a cell-by-cell basis. Cell-based discarding is the default behavior.

The PD setting applies to all discards, whether the discards occur for reasons of queue exhaustion or policing. PD is disabled by default; use the command **atm pvc** *vpi vci* **pd** {**on** | **off**} to enable or disable it.

The output queue discard thresholds work as follows:

- If packet-based discard is in force (the PD feature is enabled), the output queue absorbs packets until the queue reaches the **epd** threshold. At that point, the queue absorbs the remainder of the current packet, as long as doing so does not cause the queue to fill completely. (The total queue size equals **epd** value plus **drop** value.)

After it reaches the **epd** threshold, the queue drops all subsequent packets until the queue's contents drop below the **epd** threshold. If the queue fills completely before the current packet finishes, then PPD occurs.

- If cell-based discard is in force (the PD feature is disabled), simply add the **epd** and **drop** threshold values to determine the output queue size. When the queue is full, it drops all subsequent cells until its contents fall below the combined threshold value.

If packet-based discard is in force, you can implicitly configure the output queue discard thresholds for either EPD or PPD. For EPD, configure a **drop** threshold value that is large enough to allow most packets to enter the queue. Appropriate values for this purpose vary by traffic type, but see the thresholds in the table of defaults, below, for examples of EPD settings. For PPD, configure a very small **drop** threshold value. This forces the system to discard the remainder of the packet that fills up the queue.

The default output queue discard threshold varies by traffic priority, as shown here:

Queue Segment	cbr	vbr-rt	vbr-nrt	ubr
epd	128 cells	128 cells	1024 cells	1024 cells
drop	128 cells	128 cells	1024 cells	1024 cells
Total queue	256 cells	256 cells	2048 cells	2048 cells

To configure the output threshold, perform this task, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm output-threshold** {**cbr** | **vbr** | **abr-ubr**} {**epd** | **drop**} *disc-thresh-num*	Set the atm output epd/drop threshold to the percentage *disc-thresh-num*.

Note These commands affect all connections, including those already established. These commands do not apply to subinterface level configurations.

Example

This example shows how to configure the interface output threshold CBR epd threshold to 87 percent of maximum size and displays the results:

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm output-threshold cbr epd 87
Switch# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        epd threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
```

```
<output deleted>
```

Configuring Modem Port Input Maximum Queue Size

The input queue for local modems (subscriber ports) can be further subdivided on a per-port basis. This ensures that no one modem can consume more than its allotted share of input buffer space. This buffer space can be uniquely set for priority. It can also be uniquely set for each of the up to 256 possible modem ports. However, it can not be uniquely set for each VC. All VCs at a particular priority per port are counted against the configured value.

Note Although it is not required, it is recommended that all modem ports be configured to the same value at the same priority.

The **force** argument indicates that the change should be made even if it results in the loss of data on the interface queue. (The queue must be momentarily disabled for the threshold to be changed.) This command without the **force** argument changes only the threshold if the interface is down. An error message appears and the command does not take effect if the interface is up and the **force** argument is not present.

To display both the configured and installed values of the maximum queue size, use the **show atm interface resource** command.

To configure the modem port input maximum queue size:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the modem interface to be configured.
2.	**atm input-queue** [**force**] {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr**} **max-size** *size*	Set traffic type and the size of the modem port input queue.

Example

This example shows you how to set the modem port input maximum for CBR traffic to 5000 for modem port 4/0 and displays the results:

```
Switch(config)# interface atm 4/0
Switch(config-if)# atm input-queue force cbr max-size 5000
Switch# show atm interface resource a4/0   !(Modem Port )
Resource Management configuration:
    Output queues:
        Max sizes(derived): 256  cbr, 256  vbr-rt, 2048  vbr-nrt, 2048  ubr
        EPD  threshold: 128 cbr, 128 vbr-rt, 1024 vbr-nrt, 1024 ubr
        Drop threshold: 128 cbr, 128 vbr-rt, 1024 vbr-nrt, 1024 ubr
    Input queues:
        Max sizes(explicit cfg): 5000 cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 1024 cbr, 1024 vbr-rt, 8192 vbr-nrt, 8192 ubr
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none abr RX, none abr TX, none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
```

```
                        none abr RX, none abr TX, none ubr RX, none ubr TX
    Best effort connection limit: disabled  0 max connections
<output deleted>
```

Configuring QoS Default Values

To configure QoS default values, follow these steps:

Step	Command	Task
1.	**atm qos default** {**cbr** | **vbr-rt**} **max-cell-transfer-delay** {*microsecs* | **any**}	Selects the QoS default CBR or VBR-RT mctd.
2.	**atm qos default** {**cbr** | **vbr-rt**} **peak-to-peak-cell-delay-variation** {*microsecs* | **any**}	Selects the QoS default CBR or VBR-RT PpCDV.
3.	**atm qos default** {**cbr** | **vbr-rt** | **vbr-nrt**} **max-cell-loss-ratio** [**clp0** | **clp1plus0**] [*loss-ratio-exponent*]	Selects the QoS default CBR, VBR-RT, or VBR-NRT maximum cell loss ratio.

Note The DSLAM does not distinguish between cells that have CLP = 0 and cells that have CLP = 1+0 with respect to the policing of incoming cells. Therefore there is effectively no difference in the configuration when CLP0 or CLP1+0 is chosen. This also applies to any values that display as the result of a **show** command.

Note No traffic shaping or policing is available in the downstream direction.

The DSLAM uses these values to check the QoS parameter on the flow of a connection request, and when accumulating QoS metrics for PNNI.

Example

This example shows you how to set the QoS default to CBR, clp1plus0, with a loss-ratio exponent of 12 and displays the results:

```
DSLAM# atm qos default cbr max-cell-loss-ratio clp1plus0 12
DSLAM# show atm qos-defaults
Default QoS objective table:
    Max cell transfer delay (in microseconds): any cbr, any vbr-rt
    Peak-to-peak cell  delay variation (in microseconds): any cbr, any vbr-rt
    Max cell loss ratio for CLP0 cells: any cbr, any vbr-rt, any vbr-nrt
    Max cell loss ratio for CLP0+1 cells:  10**(-12) cbr, any vbr-rt, any vbr-nrt
```

Configuring clp-drop Setting

To enable or disable the CLP-drop flag for a service category, use this command:

Command	Task
atm clp-drop [**force**] {**vbr-nrt** | **ubr**} {**off** | **on**}	Enables or disables the CLP drop flag for the specified service category.

When the switch enables the CLP-drop flag for a service category, it drops the cells when the service-category queue reaches 50 percent of the discard threshold limit. The default is **off**.

Example

This example enables the CLP-drop flag for VBR-NRT and displays the results:

```
DSLAM# atm clp-drop force vbr-nrt on

DSLAM# show atm resource
Resource configuration:
    Over-subscription-factor 8  Sustained-cell-rate-margin-factor 1%
    Abr-mode:   relative-rate

Subtended Port Input queue Configuration:
   EPD threshold: 512 cbr, 512 vbr-rt, 4096 vbr-nrt, 4096 ubr
   Drop threshold: 512 cbr, 512 vbr-rt, 4096 vbr-nrt, 4096 ubr
   Max queue sizes (Derived):  1024  cbr,  1024  vbr-rt,  8192  vbr-nrt,  8192  ubr

CLP Drop configuration:
        cbr : off,  vbr-rt : off,  vbr-nrt : on,  ubr : off
```

Configuring the Default QoS Objective Table

Because UNI 3 signaling does not provide information elements (IEs) to signal QoS values, resource management provides a table of default objective values for QoS for guaranteed service categories. These values are used as the criteria for connection setup requirements. The values are either

- Metric values (accumulated over multiple hops of a call)
- Attributes (a gating criterion that is not accumulated, but is checked at each interface)

Maximum cell transfer delay and peak-to-peak cell delay variation are metrics, while cell loss ratio is an attribute.

Note You can configure objective values for QoS for guaranteed service categories for UNI 4.0 signaling.

Table 9-1 lists the default values of the QoS objective table.

Table 9-1 Default QoS Objective Table Row Contents

Service Category	Max Cell Transfer Delay (clp01)	Peak-to-Peak Cell Delay Variation (clp01)	Cell Loss Ratio (clp0)	Cell Loss Ratio (clp0+1)
CBR	Undefined	Undefined	Undefined	Undefined
VBR-RT	Undefined	Undefined	Undefined	Undefined

Table 9-1 Default QoS Objective Table Row Contents

Service Category	Max Cell Transfer Delay (clp01)	Peak-to-Peak Cell Delay Variation (clp01)	Cell Loss Ratio (clp0)	Cell Loss Ratio (clp0+1)
VBR-NRT	—	—	Undefined	Undefined
UBR	Undefined	Undefined	Undefined	Undefined

Note The DSLAM does not distinguish between CLP0, CLP01, and CLP0+1 with respect to the policing of incoming cells. Therefore there is effectively no difference in the configuration when CLP0, CLP01, or CLP0+1 is chosen. This also applies to any values that display as the result of a **show** command.

Note No traffic shaping or policing is available in the downstream direction.

You can assign each objective either a defined or an undefined value. If it is undefined, DSLAM does not consider the objective when it performs the connection setup.

Configure this table with the same values for an entire network.

To configure the default QoS objective table, perform these tasks in global configuration mode:

Step	Command	Task
1.	**atm qos default** {**cbr** | **vbr-rt** | **ubr**} **max-cell-transfer-delay** {*microseconds* | **any**}	Select the ATM QoS default CBR or VBR-RT maximum cell transfer delay.
2.	**atm qos default** {**cbr** | **vbr-rt** | **ubr**} **peak-to-peak-cell-delay variation** {*microseconds* | **any**}	Select the ATM QoS default CBR or VBR-RT peak-to-peak cell delay variation.
3.	**atm qos default** {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr**} **max-cell-loss-ratio** [**clp0** | **clp1plus0**] {*loss-ratio-exponent* | **any**}	Select the ATM QoS default CBR, VBR-RT, or VBR-NRT maximum cell loss ratio.

Note The DSLAM does not distinguish between cells that have CLP=0 and cells that have CLP=1 with respect to the policing of incoming cells. Therefore there is effectively no difference in the configuration when CLP0 or CLP1plus0 is chosen. This also applies to any values that display as the result of a **show** command.

Note No traffic shaping or policing is available in the downstream direction.

Example

This example shows how to change the CBR maximum cell loss ratio objective for cell loss priority (CLP) = 0 + 1 to 10^{-12} cells per second and displays the results:

```
Switch(config)# atm qos default cbr max-cell-loss-ratio clp1plus0 12

Switch# show atm qos-defaults
Default QoS objective table:
    Max cell transfer delay (in microseconds): any cbr, any vbr-rt
    Peak-to-peak cell  delay variation (in microseconds): any cbr, any vbr-rt
Max cell loss ratio for CLP0+1 cells:  10**(-12) cbr, any vbr-rt, any vbr-nrt
```

Configuring the Connection Traffic Table

You must create a row in the connection traffic table (CTT) for each unique combination of traffic parameters. Virtual path links (VPLs) and virtual channel links (VCLs) then specify traffic by specifying a row in the table per flow (receive and transmit). Several VCL/VPLs can refer to the same row in the traffic table.

CTT rows specifying these new parameters can be configured, with this effect:

- Non-zero MCR is not supported. The DSLAM rejects requests for connections specifying non-zero MCR.
- On VBR connections, the DSLAM uses only SCR and MBS for UPC.

The NI-2 module supports four traffic priorities. These four priorities are mapped to four classes of service as follows:

- Constant bit rate (CBR)—video and voice
- Variable bit rate with a remote terminal (VBR-rt)—voice
- Variable bit rate, no remote terminal (VBR-nrt)—voice
- Undefined/available bit rate (UBR/ABR)

Configuring PVC Connection Traffic Rows

The CTT in a permanent virtual channel (PVC) setup requires that you store PVC traffic values in a CTT data structure. Rows used for PVCs are called stable rows, and contain traffic parameters.

Configuring SVC Connection Traffic Rows

To configure the connection traffic table in a switched virtual circuit (SVC) setup, you create a row identifier that Simple Network Management Protocol (SNMP) or the user interface uses to read or display SVC traffic parameters. The DSLAM stores a CTT row index in the connection-leg data structure for each flow of the connection.

Note You cannot delete rows while they are in use by a connection.

To make CTT management software more efficient, DSLAM splits the CTT row-index space into

- Rows allocated as a result of signaling
- Rows allocated from the command-line interface (CLI) and SNMP

Table 9-2 describes the row-index range for both row types.

Table 9-2 CTT Row-Index Allocation

Allocated by	Row-Index Range
ATOMMIB Traffic Descriptor Table / CLI connection-traffic-table-row creation	1 through 1,073,741,823
Signaling VxL creation	1,073,741,824 through 2,147,483,647

Table 9-3 describes the CTT rows predefined by the software.

Table 9-3 Default Connection Traffic Table Rows

CTT Row Index	Service Category	Peak-Cell-Rate (CLP01)	Sustained-Cell-Rate (CLP01)	Tolerance	Use
1	UBR	7113539	—	None	Default PVP/PVC row index
2	CBR	424 kbps	—	None	CBR tunnel well-known (WK) VCs
3	VBR-RT	424 kbps	424 kbps	50%	Physical interface/VBR-RT WK VCs
4	VBR-NRT	424 kbps	424 kbps	50%	VBR-NRT tunnel WK VCs
5	UBR	424 kbps	—	None	UBR tunnel WK VCs

The **atm connection-traffic-table-row** command contains four variables—one for each service category (CBR, VBR-RT, VBR-NRT, and UBR). To create or delete a CTT row, perform these tasks in global configuration mode:

Step	Command	Task
1.	**atm connection-traffic-table-row** [**index** *row-index*] {**vbr-rt** | **vbr-nrt**} **pcr** *pcr_value* {**scr10** | **scr0**} *scr_value* [**mbs** *mbs_value*] [**cdvt** *cdvt_value*]	Configure an ATM CTT VBR row.
2.	**atm connection-traffic-table-row** [**index** *row-index*] **cbr pcr** *pcr_value* [**cdvt** *cdvt_value*]	Configure an ATM CTT CBR row.
3.	**atm connection-traffic-table-row** [**index** *row-index*] **ubr pcr** *pcr_value* [**mcr** *mcr_value*] [**cdvt** *cdvt_value*]	Configure an ATM CTT UBR row.

Note The DSLAM does not distinguish between cells that have SCR = 10 and cells that have SCR = 0 with respect to the policing of incoming cells. Therefore there is effectively no difference in the configuration when SCR10 or SCR0 is chosen. This also applies to any values that display as the result of a **show** command.

Note No traffic shaping or policing is available in the downstream direction.

These commands affect all connections, including those already established.

If you do not specify an index row number, the system software determines if one is free and displays it in the allocated index field if the command is successful.

Example

This example shows how to configure an ATM CTT row with cbr traffic and a peak cell rate of 30,000 kbps, and displays the results:

```
Switch# (config)# atm connection-traffic-table-row cbr pcr 30000
 Allocated index = 64000
Switch># show atm connection-traffic-table
Row      Service-category    pcr        scr/mcr       mbs         cdvt
1            ubr          7113539        none                      none
2            cbr             424                                  none
3            vbr-rt          424          424          50         none
4            vbr-nrt         424          424          50         none
5            ubr             424        none                      none
64000        cbr           30000        none
2147483645*  ubr                0        none                      none
2147483646*  ubr                1        none                      none
2147483647*  ubr          7113539        none                      none
```

Configuring the Sustained Cell Rate Margin Factor

The sustained cell rate margin factor determines the aggressiveness of weighting sustained cell rate (SCR) compared to peak cell rate (PCR). It uses the connection admission control algorithm in admitting VBR connections.

To configure the SCR for the DSLAM, use this global configuration command:

Command	Task
atm sustained-cell-rate-margin-factor *s-value*	Configure the sustained cell rate margin factor as a percentage.

Note The **atm sustained-cell-rate-margin-factor** command affects subsequent connections but not connections that are already established.

Example

This example shows how to configure the SCR margin factor as 64 percent of maximum and displays the results:

```
DSLAM(config)# atm sustained-cell-rate-margin-factor 64

DSLAM(config)# show atm resource
Resource configuration:
     Over-subscription-factor 8  Sustained-cell-rate-margin-factor 64%
```

```
Subtended Port Input queue Configuration:
   Max sizes: 1024  cbr, 1024  vbr-rt, 8192  vbr-nrt, 8192  ubr
   Discard threshold: 50% cbr, 50% vbr-rt, 50% vbr-nrt, 50% ubr

CLP Drop configuration:
      cbr : off,  vbr-rt : off,  vbr-nrt : off,  ubr : off
```

Configuring the Number of Best-Effort UBR Connections

To set the number of best-effort ubr connections:

Command	Task
atm cac best-effort-limit *conn-value*	Sets the number of best-effort UBR connections to *conn-value.*

Example

This example shows you how to set the number of best-effort UBR connections to 2000 and displays the results:

```
DSLAM(config)# atm cac best-effort-limit 2000

DSLAM(config)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Efci threshold: 50% cbr, 25% vbr-rt, 25% vbr-nrt, 25% ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
           Best effort connection limit: enabled  2000 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 1 pvcs,  0 svcs
```

Configuring the Maximum Value of Individual Traffic Parameters

The **atm cac** commands allow you to tune parameters used in the Connection Admission Control (CAC) functions. Table 9-4 describes the three types of parameters, which are configured per interface. Changes made to these parameters only affect subsequent connection setups.

Table 9-4 Connection Admission Configuration

Parameter	Description
controlled link sharing	Specifies the minimum and maximum bandwidth that can be allocated to guaranteed service (CBR or VBR) connections. You can specify maxima for CBR, VBR, and the aggregate of CBR and VBR. You can specify minima for CBR and VBR. These parameters, for a direction, are interrelated as follows (assuming these parameters are defined): • min(CBR) + min(VBR) <= 95% • min(CBR) <= max(CBR) <= 95% • min(VBR) <= max(VBR) <= 95% • min(CBR) <= max(AGG) <= 95% • min(VBR) <= max(AGG) <= 95% • max(CBR) <= max(AGG) <= 95% • max(VBR) <= max(AGG) <= 95%
traffic parameter *limits*	Specifies maximum traffic parameters (such as peak-cell-rate) that are allowed on VC setup. You can specify these independently by service category and traffic direction.
best-effort connection *limits*	A limit on the total number of ABR and UBR connections on the interface.

To set the maximum value of individual traffic parameters for an interface, apply these commands:

Command	Task
atm cac max-peak-cell-rate {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *rate*	Sets the maximum peak cell rate.
atm cac max-sustained-cell-rate {**receive** | **transmit**} *rate*	Sets the maximum sustained cell rate.
atm cac max-tolerance {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *cell-count*	Sets the maximum tolerance.
atm cac max-cvdt {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *rate*	Sets the maximum cvdt value.
atm cac max-mbs {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *rate*	Sets the maximum mbs value.
atm cac max-min-cell-rate {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *rate*	Sets the maximum and minimum cell rate values.

Example

This example sets a **peak-cell-rate** traffic parameter limit of 3001 kbps for ubr connections in the receive direction on the interface and displays the results:

```
Switch(config-if)# atm cac max-peak-cell-rate abr receive 3001

Switch(config-if)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Efci threshold: 50% cbr, 25% vbr-rt, 25% vbr-nrt, 25% ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, 50 cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
Best effort connection limit: enabled  2000 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, 3001 ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 1 pvcs,  0 svcs
```

Reserving Guaranteed Bandwidth for a Service Category

To fine-tune connection admission control functions on a per-interface and direction basis, use these commands:

Command	Task
atm cac link-sharing max-guaranteed-service-bandwidth {**receive** | **transmit**} *percent*	Reserves maximum guaranteed service bandwidth in the receive or transmit direction on a flow/interface basis by the amount *percent*.
atm cac link-sharing max-bandwidth {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *percent*	Reserves maximum bandwidth in the receive or transmit direction for CBR, VBR, and UBR connections on a flow/interface basis by the amount *percent*.
atm cac link-sharing min-bandwidth {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *percent*	Reserves minimum bandwidth in the receive or transmit direction for CBR, VBR, and UBR connections on a flow/interface basis by the amount *percent*.

Display the Resource Management Configuration

To display the resource management configuration, use this EXEC command:

Command	Task
show atm interface resource 0/1	Display the resource management configuration.

Example

This example reserves 50 percent as the minimum bandwidth for CBR in the transmit direction and displays the results:

```
DSLAM(config)# atm cac link-sharing min-bandwidth cbr transmit 50

DSLAM(config)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Efci threshold: 50% cbr, 25% vbr-rt, 25% vbr-nrt, 25% ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, 50 cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
Best effort connection limit: enabled  2000 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
```

```
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 1 pvcs,  0 svcs
```

Configuring the Allowed Service Categories

To set the service categories CAC allows on an interface, use these commands:

Step	Command	Task
1.	`DSLAM(config)#` **interface atm 0/1**	Select interface 0/1.
2.	`DSLAM(config)#` **atm cac service-category** {**cbr** | **vbr** | **...**} {**deny** | **permit**}	Sets a service category CAC allows or denies on an interface.

Example

This example prohibits the service category cbr on the interface 0/1 and displays the results:

```
DSLAM(config)# interface atm 0/1
DSLAM(config)# atm cac service-category cbr deny
DSLAM# show running-config
Building configuration...
Current configuration:
!
! No configuration change since last restart
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname ni2-3
!
enable password lab
dmt-profile default
network-clock-select 1 ATM0/1
network-clock-select 2 system
ip subnet-zero
ip host-routing
ip domain-name cisco.com
ip name-server 171.69.204.11
!
atm address 47.0091.8100.0000.007b.f444.7801.007b.f444.7801.00
atm router pnni
```

```
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
clock timezone EST -5
clock summer-time EDT recurring
!
process-max-time 200
!
interface ATM0/0
 ip address 70.0.0.2 255.0.0.0
 no ip directed-broadcast
 map-group test
 atm cac service-category abr deny
 atm maxvp-number 0
!
interface Ethernet0/0
<output deleted>
```

Configuring the Propagation Delay (Link Distance)

You can increase the propagation delay by specifying the link distance for the physical link of the next ATM host in the outbound direction.

To set the link distance, use these commands:

Step	Command	Task
1.	`DSLAM(config)#` **interface atm 0/1**	Select interface 0/1.
2.	`DSLAM(config)#` **atm link-distance** *distance*	Sets the link distance to the value *distance*.

Example

This example shows you how to set the link distance to 64 km and displays the results:

```
DSLAM(config)# atm link-distance 64
DSLAM# show atm interface resource atm 0/1
Resource Management configuration:
         Service Classes:
        Service Category map: c1 cbr, c2 vbr-rt, c3 vbr-nrt, c4 c5 ubr
        Scheduling: RS c1 WRR c2, WRR c3, WRR c4, WRR c5
        WRR Weight: 8 c2, 1 c3, 1 c4, 1 c5
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
     Link Distance: 64 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX,
        Tolerance RX: none cbr, none vbr, none ubr
        Tolerance TX: none cbr, none vbr, none ubr
Resource Management state:
    Available bit rates (in Kbps):
        147743 cbr RX, 147743 cbr TX, 147743 vbr RX, 147743 vbr TX,
        147743 ubr RX, 147743 ubr TX
```

```
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 0 pvcs,  0 svcs
```

Configuring a CDVT and MBS Default

When the DSLAM does not specify a CDVT or MBS for PVCs through a connection traffic row, the software applies a per-interface, per-service-category default for UPC on the connection.

To set a per-interface, per-service-category CDVT or MBS default for UPC on a connection, use these commands:

Step	Command	Task
1.	DSLAM(config)# **atm cdvt-default** {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr**} *num*	Sets the CDVT default for cbr, vbr-rt, vbr-nrt, or ubr to *num*.
2.	DSLAM(config)# **atm mbs-default** {**vbr-rt** | **vbr-nrt**} *num*	Sets the MBS default for vbr-rt or vbr-nrt to *num*.

Example

This example shows you how to set the MBS vbr-rt UPC default to 20 on interface 0/1 and displays the result:

```
DSLAM(config)# interface atm 0/1
DSLAM(config)# atm mbs-default vbr-rt 20

DSLAM(config)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none aubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
         Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
            MBS: none vbr RX, none vbr TX, 20 vbr-rt
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, 0 ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
```

```
Allocated bit rates:
    0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
    0 ubr RX, 0 ubr TX
Best effort connections: 1 pvcs,  0 svcs
```

Configuring CAC Functions for Specific Interfaces and Directions

Resource management lets you fine-tune the connection admission control functions on a per-interface and direction (receive and transmit) basis. You specify the reservations with these parameters:

- Maximum aggregate guaranteed cell rate on an interface, which limits the guaranteed bandwidth that can be allocated on an interface
- Maximum guaranteed cell rates on an interface per-service category
- Minimum guaranteed cell rates on an interface per-service category

The connection admission control parameter to bandwidth relationships are shown in Table 9-5.

Table 9-5 Connection Admission Control Parameter to Bandwidth Relationships

Service Category	Value	Service Category	Bandwidth (Percent)
Minimum CBR	+	Minimum VBR	<= 95
Minimum CBR	<=	Maximum CBR	<= 95
Minimum VBR	<=	Maximum VBR	<= 95
Minimum CBR	<=	Maximum Aggregate	<= 95
Minimum VBR	<=	Maximum Aggregate	<= 95
Maximum CBR	<=	Maximum Aggregate	<= 95
Maximum VBR	<=	Maximum Aggregate	<= 95

To configure controlled link sharing, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	`DSLAM(config)#` **interface atm** *slot/port* [*.vpt#*]	Select the interface to be configured.
2.	`DSLAM(config)#` **atm cac link-sharing max-guaranteed-service-bandwidth** {**receive** | **transmit**} *percent*	Configure controlled link sharing for the maximum guaranteed service bandwidth.

Step	Command	Task
3.	DSLAM(config)# **atm cac link-sharing max-bandwidth** {**cbr** | **ubr** | **vbr**} {**receive** | **transmit**} *percent*	Configure controlled link sharing for the maximum guaranteed service bandwidth by service category.
4.	DSLAM(config)# **atm cac link-sharing min-bandwidth** {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *percent*	Configure controlled link sharing for the minimum guaranteed service bandwidth by service category.

Note These commands affect subsequent connections, but not connections that are already established.

For restrictions to these commands, see the *Command Reference for Cisco DSLAMs with NI-2*.

Examples

This example shows how to configure the controlled link sharing, maximum guaranteed service bandwidth, and receive configuration on interface 0/1 to 87 percent and displays the result:

```
DSLAM(config)# interface atm 0/1
DSLAM(config)# atm cac link-sharing max-guaranteed-service-bandwidth receive 87
DSLAM(config)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
    Best effort connection limit: disabled  0 max connections

<output deleted>
```

This example displays the controlled link sharing configuration:

```
DSLAM(config)# show atm interface resource atm 0/0
Resource Management configuration:
    Service Classes:
        Service Category map: c1 cbr, c2 vbr-rt, c3 vbr-nrt, c5 ubr
        Scheduling: RS c1 WRR c2, WRR c3, WRR c4, WRR c5
        WRR Weight: 8 c2, 1 c3, 1 c4, 1 c5
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX,
```

```
        Tolerance RX: none cbr, none vbr, none ubr
        Tolerance TX: none cbr, none vbr, none ubr
Resource Management state:
    Available bit rates (in Kbps):
        147743 cbr RX, 147743 cbr TX, 147743 vbr RX, 147743 vbr TX,
        147743 ubr RX, 147743 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 0 pvcs,  0 svcs
```

Configuring the Physical and Logical Interface Parameters

This section describes interface configuration resource management tasks for both physical and logical interface types.

Configuring the Outbound Link Distance

Specifying the physical link distance for the next ATM hop in the outbound direction allows you to increase the propagation delay. DSLAM uses the propagation delay to determine the connection admission control (CAC) maximum CTD provided on the output by a switch interface, which can affect the SVC connection requests accepted.

To configure the ATM link distance, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM(config)# **interface atm** *slot/port*[*.vpt#*]	Select the subinterface to be configured.
2.	DSLAM(config)# **atm link-distance** *kilometers*	Configure the subinterface link distance.

Note The **atm link-distance** command affects subsequent connections, but not connections that are already established.

Examples

This example shows how to configure the interface link distance configuration to 150 kilometers and displays the result:

```
DSLAM(config)# atm link-distance 150

DSLAM(config)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Efci threshold: 50% cbr, 25% vbr-rt, 25% vbr-nrt, 25% ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt,87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
            Link Distance: 150 kilometers
```

```
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr,none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX:none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 1 pvcs,  0 svcs
```

This example shows you how to set configuration of the interface link distance:

```
DSLAM(config)# show atm interface resource atm 0/0
Resource Management configuration:
    Service Classes:
        Service Category map: c1 cbr, c2 vbr-rt, c3 vbr-nrt, c5 ubr
        Scheduling: RS c1 WRR c2, WRR c3, WRR c4, WRR c5
        WRR Weight: 8 c2, 1 c3, 1 c4, 1 c5
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX,
        Tolerance RX: none cbr, none vbr, none ubr
        Tolerance TX: none cbr, none vbr, none ubr
Resource Management state:
    Available bit rates (in Kbps):
        147743 cbr RX, 147743 cbr TX, 147743 vbr RX, 147743 vbr TX,
        147743 ubr RX, 147743 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 0 pvcs,  0 svcs
```

Configuring the Limits of Best-Effort Connections

You can configure each interface to allow a specific number of best-effort UBR connections.

To configure the number of best-effort connections, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM(config)# **interface atm** *slot/port*[*.vpt#*]	Select the interface to be configured.
2.	DSLAM(config-if)# **atm cac best-effort-limit** *conn-value*	Configure the connection best-effort limit.

Note These commands affect subsequent connections but not connections that are already established.

Example

This example configures the connection best-effort limit configuration for interface 0/1 to 2000 and displays the result:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm cac best-effort-limit 2000

DSLAM(config)# show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
    Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt,ubr
    Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
    Best effort connection limit: enabled  2000 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, 0 ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 1 pvcs,  0 svcs
```

This example shows the interface best-effort configuration disabled:

```
Switch# show atm interface resource atm 0/0
Resource Management configuration:
    Service Classes:
        Service Category map: c1 cbr, c2 vbr-rt, c3 vbr-nrt, c5 ubr
        Scheduling: RS c1 WRR c2, WRR c3, WRR c4, WRR c5
        WRR Weight: 8 c2, 1 c3, 1 c4, 1 c5
```

```
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX,
        Tolerance RX: none cbr, none vbr, none ubr
        Tolerance TX: none cbr, none vbr, none ubr
Resource Management state:
    Available bit rates (in Kbps):
        147743 cbr RX, 147743 cbr TX, 147743 vbr RX, 147743 vbr TX,
        147743 ubr RX, 147743 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 0 pvcs,  0 svcs
```

Configuring the Interface Maximum of Individual Traffic Parameters

When a VCC is set up, you can specify per-flow (receive and transmit traffic) parameters. You can configure traffic parameter limits independently (by service category), as well as traffic direction for

- Maximum peak cell rate (PCR)
- Maximum sustained cell rate (SCR)
- Maximum cell delay variation tolerance (CDVT)
- Maximum burst size (MBS)
- Maximum or minimum cell rate (MCR)

To configure the traffic parameters, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.vpt#*]	Select the interface to be configured.
2.	**atm cac max-peak-cell-rate** {**cbr** | **vbr** | **ubr**} {**receive** | **transmit**} *rate*	Configure the connection maximum PCR.
3.	**atm cac max-sustained-cell-rate** {**receive** | **transmit**} *rate*	Configure the connection maximum SCR.
4.	**atm cac max-cdvt** {**cbr** | **ubr** | **vbr**} {**receive** | **transmit**} *cell-count*	Configure the connection maximum CDVT.
5.	**atm cac max-mbs** {**receive** | **transmit**} *cell-count*	Configure the connection maximum MBS.
6.	**atm cac max-min-cell-rate** {**ubr**} {**receive** | **transmit**} *rate*	Configure the connection MCR per service category flow.

Note These commands affect subsequent connections but not connections that are already established.

Examples

This example shows how to configure the maximum PCR for CBR connections on interface 0/1, specified in receive mode, to 100,000 kbps:

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm cac max-peak-cell-rate cbr receive 100000
```

This example shows how to configure the maximum SCR for connections on interface 0/1, specified in receive mode, to 60,000 kbps:

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm cac max-sustained-cell-rate receive 60000
```

This example shows how to configure the maximum tolerance for CBR connections on interface 0/1, specified in receive mode, 75,000 kbps:

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm cac max-cdvt cbr receive 75000
```

This example shows the interface output pacing configuration for interface 0/1:

```
Switch> show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): 30000  cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 30208 cbr, 256 vbr-rt, 4096 vbr-nrt, 12032 ubr
        Efci threshold: 50% cbr, 25% vbr-rt, 25% vbr-nrt, 25% ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: enabled   10000 Kbps rate configured, 10000 Kbps rate installed
    Service Categories supported: cbr,vbr-rt,vbr-nrt, ubr
    Link Distance: 150 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: 87% Rx,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
    Best effort connection limit: enabled  2000 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: 60000 vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: 75000 cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, 0 ubr
    Available bit rates (in Kbps):
        135302 cbr RX, 9499 cbr TX, 135302 vbr RX, 9499 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 1 pvcs,  0 svcs
```

This example shows the interface output pacing configuration for the interface 0/0:

```
Switch# show atm interface resource atm 0/0
Resource Management configuration:
    Service Classes:
        Service Category map: c1 cbr, c2 vbr-rt, c3 vbr-nrt, c5 ubr
        Scheduling: RS c1 WRR c2, WRR c3, WRR c4, WRR c5
        WRR Weight: 8 c2, 1 c3, 1 c4, 1 c5
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX,
        Tolerance RX: none cbr, none vbr, none ubr
        Tolerance TX: none cbr, none vbr, none ubr
Resource Management state:
    Available bit rates (in Kbps):
        147743 cbr RX, 147743 cbr TX, 147743 vbr RX, 147743 vbr TX,
        147743 ubr RX, 147743 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 0 pvcs,  0 svcs
```

Configuring the ATM Default CDVT and MBS

You can change the default cell delay variation tolerance (CDVT) and maximum burst size (MBS) to request for UPC of cells received on the interface for connections that do not individually request a CDVT or MBS value. To do so, use the **atm cdvt-default** or **atm mbs-default** interface configuration commands. To reset the default CDVT for a particular service category to the default value, use the **no** form of this command.

You can specify CDVT or MBS for PVCs using a connection traffic table row. If no CDVT or MBS is specified in the row, then a per-interface, per-service category default is applied for purposes of UPC on the connection.

Note For signaled connections, you cannot use CDVT or MBS and the defaults specified on the interface apply.

To configure the default CDVT and MBS parameters, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.

Step	Command	Task
2.	**atm cdvt-default** {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr**} *num*	Configure the ATM CDVT default.
3.	**atm mbs-default** {**vbr-rt** | **vbr-nrt**} *num*	Configure the ATM MBS default.

Example

This example shows how to change the default tolerance for received cells on VBR-RT connections:

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm cdvt-default vbr-rt 4000
```

Display the ATM CDVT and MBS Configuration

To display the ATM CDVT and MBS configuration, use these EXEC commands:

Command	Task
show atm vc	Display the ATM VC CDVT configuration.
show atm vp	Display the ATM VP CDVT configuration.

Examples

This example shows the ATM CDVT and MBS configuration of an ATM VC for interface 0/1, with VPI = 0:

```
Switch# show atm vc interface atm 0/1 0 100

Interface: ATM0/1, Type: suni-dual
VPI = 0  VCI = 100
Status: UP
Time-since-last-status-change: 00:02:51
Connection-type: PVC
Cast-type: point-to-point
Packet-discard-option: disabled
Usage-Parameter-Control (UPC): drop
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/0, Type: suni-dual
Cross-connect-VPI = 0
Cross-connect-VCI = 100
Cross-connect-UPC: drop
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Rx cells: 0, Tx cells: 0
Rx connection-traffic-table-index: 80001
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 80000
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 100
Rx       mbs: none
Tx connection-traffic-table-index: 80001
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 80000
```

```
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: 100
Tx       mbs: none
```

This example shows the ATM CDVT and MBS configuration of an ATM VP for interface 0/1, with VPI = 4:

```
Switch# show atm vp interface atm 0/1 4

Interface: ATM0/1, Type: suni-dual
VPI = 4
Status: UP
Time-since-last-status-change: 00:00:11
Connection-type: PVP
Cast-type: point-to-point
Usage-Parameter-Control (UPC): pass
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/1, Type: suni-dual
Cross-connect-VPI = 4
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Rx cells: 0, Tx cells: 0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

This example shows the ATM CDVT and MBS configuration of an ATM VP for interface 0/1, with VPI = 4:

```
Switch# show atm vp interface atm0/1 4
Interface: ATM0/1, Type: suni-dual
VPI = 4
Status: UP
Time-since-last-status-change: 00:00:10
Connection-type: PVP
Cast-type: point-to-point
Usage-Parameter-Control (UPC): pass
Wrr weight: 32
Number of OAM-configured connections: 0
OAM-configuration: disabled
OAM-states:  Not-applicable
Cross-connect-interface: ATM0/2, Type: suni-dual
Cross-connect-VPI = 4
Cross-connect-UPC: pass
Cross-connect OAM-configuration: disabled
Cross-connect OAM-state:  Not-applicable
Threshold Group: 5, Cells queued: 0
Rx cells: 0, Tx cells: 0
Tx Clp0:0,  Tx Clp1: 0
Rx Clp0:0,  Rx Clp1: 0
Rx Upc Violations:0, Rx cell drops:0
```

```
Rx Clp0 q full drops:0, Rx Clp1 qthresh drops:0
Rx connection-traffic-table-index: 1
Rx service-category: UBR (Unspecified Bit Rate)
Rx pcr-clp01: 7113539
Rx scr-clp01: none
Rx mcr-clp01: none
Rx      cdvt: 1024 (from default for interface)
Rx       mbs: none
Tx connection-traffic-table-index: 1
Tx service-category: UBR (Unspecified Bit Rate)
Tx pcr-clp01: 7113539
Tx scr-clp01: none
Tx mcr-clp01: none
Tx      cdvt: none
Tx       mbs: none
```

Configuring Interface Service Category Support

With interface service category support, you can configure the service categories that CAC allows on an interface. You can configure interface service category support only on physical interfaces and shaped VP tunnel logical interfaces. The underlying VP for shaped VP tunnel logical interfaces must use the CBR service category. By default, only CBR user VCs can cross the interface. Using the service category support configuration commands, you can substitute another service category for CBR on the interface. This configuration ensures that your switch shapes traffic according to the aggregate VP traffic contract before it enters a service provider network.

Note No traffic shaping or policing is available in the downstream direction.

Table 9-6 shows the service category of the shaped VP (always CBR), the service categories you can configure for transported VCs, and a suggested transit VP service category for the tunnel.

Table 9-6 Service Category Support for Shaped VP Tunnels

Shaped VP Tunnel Service Category	VC Service Category	Suggested Transit VP Service Category
CBR	CBR	CBR
CBR	VBR	CBR or VBR
CBR	UBR	Any service category

These restrictions apply to interface service category support:

- This configuration is allowed on physical interfaces and shaped VP tunnel logical interfaces.
- On shaped VP tunnel logical interfaces, only one service category is permitted at a time. To replace CBR with another service category on these interfaces, you must first deny the CBR service category, then permit the chosen service category. To deny a service category, you must delete all user VCs of that service category on the interface.
- For UBR, only zero MCR is supported on VCs on a shaped VP tunnel.

To configure a service category on an interface, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.vpt#*]	Select the interface to be configured.
2.	**atm cac service-category** {**cbr** | **vbr-rt** | **vbr-nrt** | **abr** | **ubr**} {**permit** | **deny**}	Configure the service category on the interface.

Note **abr** appears in this command syntax, but the current release does not support it.

Example

This example shows how to deny the UBR service category on ATM interface 0/1 and displays the result:

```
Switch(config)# interface atm 0/1
Switch(config-if)# atm cac service-category ubr deny

Switch> show atm interface resource atm 0/1
Resource Management configuration:
    Output queues:
        Max sizes(explicit cfg): none cbr, none vbr-rt, none vbr-nrt, none ubr
        Max sizes(installed): 256 cbr, 512 vbr-rt, 4096 vbr-nrt, 11776 ubr
        Discard threshold: 87% cbr, 87% vbr-rt, 87% vbr-nrt, 87% ubr
    Pacing: disabled   0 Kbps rate configured, 0 Kbps rate installed
     Service Categories supported: cbr,vbr-rt,vbr-nrt
    Link Distance: 0 kilometers
    Controlled Link sharing:
        Max aggregate guaranteed services: none RX,  none TX
        Max bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
        Min bandwidth: none cbr RX, none cbr TX, none vbr RX, none vbr TX,
                       none ubr RX, none ubr TX
    Best effort connection limit: disabled  0 max connections
    Max traffic parameters by service (rate in Kbps, tolerance in cell-times):
        Peak-cell-rate RX: none cbr, none vbr, none ubr
        Peak-cell-rate TX: none cbr, none vbr, none ubr
        Sustained-cell-rate: none vbr RX, none vbr TX
        Minimum-cell-rate RX: none ubr
        Minimum-cell-rate TX: none ubr
        CDVT RX: none cbr, none vbr, none ubr
        CDVT TX: none cbr, none vbr, none ubr
        MBS: none vbr RX, none vbr TX
Resource Management state:
    Cell-counts: 0 cbr, 0 vbr-rt, 0 vbr-nrt, 0 ubr
    Available bit rates (in Kbps):
        1466 cbr RX, 1466 cbr TX, 1466 vbr RX, 1466 vbr TX,
        0 ubr RX, 0 ubr TX
    Allocated bit rates:
        0 cbr RX, 0 cbr TX, 0 vbr RX, 0 vbr TX,
        0 ubr RX, 0 ubr TX
    Best effort connections: 0 pvcs,  0 svcs
```

CHAPTER 10

Configuring ILMI

This chapter describes the Integrated Local Management Interface (ILMI) protocol implementation for Cisco DSLAMs with NI-2 cards, and includes the sections:

- ILMI Overview
- Configuring the Global ILMI System
- Configuring an ILMI Interface

ILMI Overview

The DSLAM uses ILMI to automatically identify which of its interfaces are User-Network Interface (UNI), attached to ATM end systems, and which are Network-to-Network Interface (NNI), attached to other systems. It can also differentiate between private and public network links. This information is used by the ATM routing protocols, Private Network-to-Network Interface (PNNI), and Interim-Interswitch Signaling Protocol (IISP) to automatically discover and gain access to a network of interconnected DSLAMs.

The ILMI protocol is also used for ATM address registration across an ATM UNI, and for

- Configuring ATM end systems with ATM address prefixes
- Allowing the DSLAM to discover the 48-bit Media Access Control (MAC) addresses of the attached systems

ILMI reduces the need for manual configuration of attached end systems and is important in the operation of DSLAM-based networks.

Configuring the Global ILMI System

This section describes configuring the ATM address and displaying the ILMI configuration for the entire DSLAM.

Configuring the ATM Address

The DSLAM is autoconfigured with an ATM address using a hierarchical addressing model similar to the Open System Interconnection (OSI) network service access point (NSAP) addresses. PNNI uses this hierarchy to construct ATM peer groups. ILMI uses the first 13 bytes of this address as the switch prefix that it registers with end systems.

During the initial startup, the DSLAM generates an ATM address using the defaults described in Chapter 3, "Initially Configuring the Cisco DSLAM."

Note The most important rule in the addressing scheme is to maintain the uniqueness of the address across very large networks.

To configure a new ATM address that replaces the previous ATM address and generates a new PNNI node ID and peer group ID, see Chapter 11, "Configuring ATM Routing and PNNI."

You can configure multiple addresses for a single DSLAM, and use this configuration during ATM address migration. ILMI registers end systems with multiple prefixes during this period until an old address is removed. PNNI automatically summarizes all of the switch's (or DSLAM's) prefixes in its reachable address advertisement.

To configure a new ATM address that replaces the previous ATM address when you are running IISP only, perform these tasks in global configuration mode:

Step	Command	Task
1.	**atm address** *atm_address*	At the configuration mode prompt, configure the new ATM address for the DSLAM.
2.	**no atm address** *atm_address*	At the configuration mode prompt, remove the old ATM address from the DSLAM.

Example

This example shows how to add the ATM address prefix 47.0091.8100.5670.000.0ca7.ce01 using the ellipses (...) to add the default MAC address as the last six bytes, and displays the results.

```
DSLAM(config)# atm address 47.0091.8100.5670.0000.0ca7.ce01...
Building configuration...
Current configuration:

<information deleted>

!
atm abr-mode efci
atm lecs-address-default 47.0091.8100.0000.0040.0b0a.1281.0040.0b4e.d023.00 1
atm lecs-address-default 47.0091.8100.0000.0040.0b0a.1281.0040.0b07.4023.00 2
atm ilmi default-access permit matching-prefix
atm address 47.0091.8100.5670.000.0ca7.ce01.2b81.00.7901.00
atm address 47.0091.8100.0000.0060.3e5a.7901.0060.3e5a.7901.00
atm router pnni
 statistics call
 node 1 level 56 lowest
```

Configuring Global ILMI Access Filters

The ILMI access filter feature allows you to permit or deny certain ILMI registered addresses.

Note If you want to allow certain addresses to be registered through ILMI, but restrict those addressees from being advertised through PNNI, use the PNNI suppressed summary address feature instead. For additional information, see Chapter 11, "Configuring ATM Routing and PNNI."

If end systems are allowed to register arbitrary addresses via ILMI, including addresses that do not match the ILMI prefixes used on the interface, a security hole may be opened. The ILMI access filter feature closes the security hole by permitting or denying ILMI registration of different classes of addresses.

The ILMI access filter allows you to configure two levels of access filters:

- Globally, to configure the switch default access filter
- At the interface level, to set the per-interface specific override

In either level, you can choose among these options:

- Permit all—Any ATM end system address (AESA) registered by an attached end system is permitted.
- Permit prefix match—Only AESAs that match an ILMI prefix used on the interface are permitted. Permit prefix match, well-known group addresses assigned by the ATM Forum, and AESAs that match an ILMI prefix used on the interface are permitted.
- Permit prefix match and all group addresses—All group addresses, including the well-known group addresses, as well as AESAs that match the ILMI prefix(es) used on the interface are permitted.

To configure global ILMI access filters, use this global configuration command:

Command	Task
atm ilmi default-access permit {**all** | **matching-prefix** [**all-groups** | **wellknown-groups**]}	Configure an ILMI default access filter.

Example

This example shows how to configure the global default access filter for ILMI address registration to allow well-known group addresses and addresses with matching prefixes and displays the result:

```
DSLAM(config)# atm ilmi default-access permit matching-prefix wellknown-groups

DSLAM# show running-config
Building configuration...
Current configuration:
    <information deleted>
atm abr-mode efci
atm lecs-address-default 47.0091.8100.0000.0040.0b0a.1281.0040.0b4e.d023.00 1
atm lecs-address-default 47.0091.8100.0000.0040.0b0a.1281.0040.0b07.4023.00 2
atm ilmi default-access permit matching-prefix
atm address 47.0091.8100.5670.000.0ca7.ce01.2b81.00.7901.00
atm address 47.0091.8100.0000.0060.3e5a.7901.0060.3e5a.7901.00
atm router pnni
 statistics call
 node 1 level 56 lowest
```

Displaying the ILMI Global Configuration

To display the DSLAM ILMI configuration, use these EXEC commands:

Command	Task
show atm address	Display the ATM addresses.
show atm ilmi-configuration	Display the ILMI configuration.
show atm ilmi-status	Display the ILMI status.

Examples

This example shows the ATM address and the LECS address:

```
ni2-3# show atm addresses

Switch Address(es):
  47.009181000000007BF4447801.007BF4447801.00 active

Soft VC Address(es):
  47.0091.8100.0000.007b.f444.7801.4000.0c80.0010.00 ATM0/1
  47.0091.8100.0000.007b.f444.7801.4000.0c80.0020.00 ATM0/2

ILMI Switch Prefix(es):
  47.0091.8100.0000.007b.f444.7801

ILMI Configured Interface Prefix(es):

LECS Address(es):
```

This example shows the ILMI configuration:

```
DSLAM# show atm ilmi-configuration

Switch ATM Address (s) :
11223344556677889901122334455667788990000
LECS Address (s):
1122334455667788990011223344556677889900
ARP Server Address (s):
1122334455667788990011223344556677889900
```

This example shows the ILMI status:

```
DSLAM# show atm ilmi-status

Interface : ATM0/0 Interface Type : Local
Configured Prefix(s) :
47.0091.8100.0000.0003.c386.b301

Interface : ATM0/1 Interface Type : Private NNI
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
Configured Prefix(s) :
47.0091.8100.0000.0003.c386.b301

Interface : ATM0/2 Interface Type : Private NNI
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
Configured Prefix(s) :
47.0091.8100.0000.0003.c386.b301
```

Configuring an ILMI Interface

To configure an ILMI interface on a per-interface basis, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.vpt#*]	Select interface port.
2.	**atm auto-configuration**	Enable ILMI auto configuration, including determination of interface protocol, version and side.
3.	**atm address-registration**	Configure ILMI address registration for a specified interface.
4.	**auto-link-determination**	Enable ILMI link determination feature.
5.	**atm ilmi-keepalive** [**seconds** [**retry** *retry_number*]]	Configure ILMI keepalive.

Note If the ILMI VC (by default VCI = 16) is disabled, then the ILMI is disabled.

Examples

This example shows how to enable ILMI autoconfiguration on ATM interface 0/2:

```
DSLAM(config)# interface atm 0/2
DSLAM(config-if)# atm-autoconfiguration
```

This example shows how to enable ATM address registration on ATM interface 0/2:

```
DSLAM(config)# interface atm 0/2
DSLAM(config-if)# atm address-registration
```

Note If you use the **no atm address-registration** command to disable ILMI on this interface, the keepalives and responses to incoming ILMI queries continue to function. To disable ILMI at this interface, use the **no atm ilmi-enable** command.

This example shows how to configure the ILMI ATM interface 0/2 with a keepalive time of 20 seconds and retry count of 3 and displays the ILMI interface configuration:

```
DSLAM(config)# interface atm 0/2
DSLAM(config-if)# atm ilmi-keepalive 20 retry 3

DSLAM# show atm ilmi-status atm 0/2

Interface : ATM0/2 Interface Type : Private NNI
ILMI VCC : (0, 16) ILMI Keepalive : 20
Configured Prefix(s) :
47.0091.8100.0000.0003.c386.b301
```

Configuring per-Interface ILMI Address Prefixes

The DSLAM allows configuration of per-interface ILMI address prefixes to allow you to register different address prefixes with end systems attached to different interfaces. When you configure per-interface ILMI address prefixes, they override the prefixes derived from the first 13 bytes of the switch ATM addresses for that specific interface.

You can configure multiple ILMI address prefixes on each interface. For example, during ATM address migration.

To configure a per-interface ILMI address prefix, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.vpt#*]	Specify an ATM interface and enter interface configuration mode.
2.	**atm prefix** *13-byte-prefix*	Configure the ILMI address prefix.

Examples

This example shows how to change the ATM address of the DSLAM from the autoconfigured address 47.0091.8100.0000.0041.0b0a.1081.0041.0b0a.1081.00 to the new address 47.0091.8100.5670.0000.0000.1122.0041.0b0a.1081.00:

```
DSLAM(config)# atm address 47.0091.8100.5670.0000.0000.1122...
DSLAM(config)# no atm address 47.0091.8100.0000.0041.0b0a.1081...
```

This example shows how to configure an additional ATM addresses manually. The address prefix 47.0091.8100.0000.0003.c386.b301 on ATM interface 0/1 is configured:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm prefix 47.0091.8100.0000.0003.c386.b301
```

This example uses the **show atm address** command to display the ILMI address prefix configuration for all interfaces:

```
ni2-3# show atm addresses

Switch Address(es):
  47.009181000000007BF4447801.007BF4447801.00 active

Soft VC Address(es):
  47.0091.8100.0000.007b.f444.7801.4000.0c80.0010.00 ATM0/1
  47.0091.8100.0000.007b.f444.7801.4000.0c80.0020.00 ATM0/2

ILMI Switch Prefix(es):
  47.0091.8100.0000.007b.f444.7801

ILMI Configured Interface Prefix(es):

LECS Address(es):
```

This example displays the ILMI status for ATM interface 0/1:

```
DSLAM# show atm ilmi-status atm 0/1

Interface : ATM0/1 Interface Type : Private NNI
```

```
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
Configured Prefix(s) :
47.0091.8100.0000.0003.c386.b301
```

CHAPTER 11

Configuring ATM Routing and PNNI

This chapter describes the Interim Interswitch Signaling Protocol (IISP) and Private Network-Network Interface (PNNI) ATM routing protocol implementations on Cisco DSLAMs with NI-2. This chapter includes

ATM Routing Overview

To place calls between ATM end systems, signaling consults an IISP, a static routing protocol, or PNNI. PNNI is a dynamic routing protocol that provides quality of service (QoS) routes to signaling based on the QoS requirements specified in the call setup request.

This section provides an overview of PNNI with a comparison to IISP.

Dynamic Routing

PNNI is a dynamic routing protocol for ATM. PNNI is dynamic because it learns the network topology and reachability information with minimal configuration. It automatically adapts to network changes by advertising topology state information.

In contrast, IISP is a static routing protocol. You must manually configure each route through the network. Because IISP static routing requires significant manual configuration and does not offer the scalability of PNNI hierarchy, it is best suited for use in small networks.

Source Routing

In a PNNI routing domain, the source ATM switch (or DSLAM) computes hierarchically complete routes for connection setups. This route information is included in the call setup signaling message.

In contrast, IISP uses hop-by-hop routing, where each switch or DSLAM that receives the connection setup message selects the next outgoing interface to which to forward the setup message. This selection is based on the mapping of destination addresses (in a routing table) to outgoing interfaces.

QoS Support

PNNI provides routes that satisfy quality of service (QoS) connection requests. PNNI selects routes through the network based on the administrative weight (AW) and other QoS parameters, such as

- Available cell rate (AvCR)
- Maximum cell transfer delay (maxCTD)
- Peak-to-peak cell delay variation (CDV)
- Cell loss ratio (CLR)

The primary metric used by PNNI is AW. If a connection requests either maxCTD or CDV or both, PNNI may not be able to compute an optimum route through the network. However, PNNI guarantees a route that meets or exceeds the criteria of all specified QoS parameters.

In contrast, IISP does not provide QoS support.

PNNI Hierarchy

The primary goal of the PNNI hierarchy is scalability. However, you can also use the PNNI hierarchy for other needs, such as creating an administrative boundary. For example, you can use the PNNI hierarchy to hide the internal details of a peer group from switches outside of the peer group.

The key components of the PNNI hierarchy are:

- Lowest-level nodes—A logical node in the lowest level of the PNNI hierarchy.
- Peer group—A group of logical nodes. Each node exchanges information with other members of the group, and all members maintain an identical view of the group.
- Peer group leader (PGL)—A logical node within a peer group that summarizes the peer group and represents it as a single logical node at the next level of the PNNI hierarchy.
- Logical group node (LGN)—A logical node that represents its lower level peer group in the next higher level peer group. Upon becoming a PGL, the PGL creates a parent LGN to represent the peer group as a single logical node at the next level. The PGL is a logical node within the peer group, and the associated LGN is a logical node in the next higher level peer group.

The lowest level of the PNNI hierarchy contains lowest-level nodes only. No higher levels are possible if all nodes within a peer group are configured as lowest-level nodes. If your network is relatively small and scalability is not a problem, and the PNNI hierarchy is not required for other reasons, the benefits of a flat PNNI network may far outweigh the benefits of a hierarchical PNNI network. Refer to the "Configuring the Lowest Level of the PNNI Hierarchy" section on page 11-205 for more information.

The peer group, PGL, and LGN define the hierarchy and are needed to create multiple levels of the PNNI hierarchy. Refer to the "Configuring Higher Levels of the PNNI Hierarchy" section on page 11-211 for more information.

Figure 11-1 shows a flat network topology, where every node maintains information about every physical link in the network and reachability information for every other node in the network.

Figure 11-1 Flat Network Topology

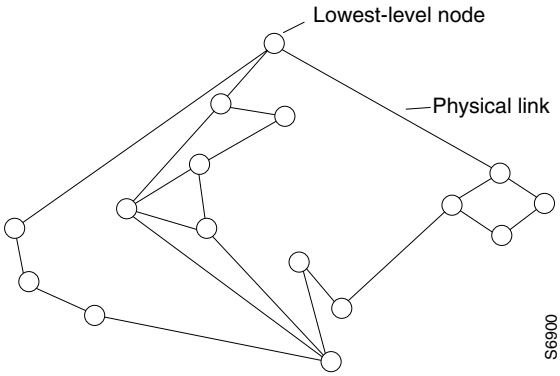
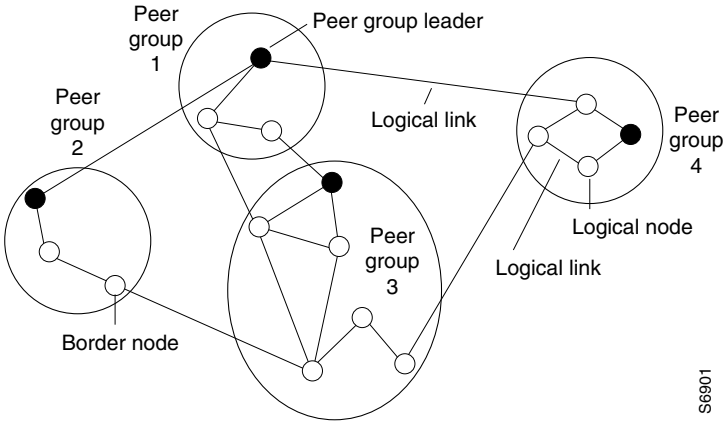


Figure 11-2 shows a PNNI hierarchical network topology. In a PNNI hierarchical network, the number of nodes, links, and reachable address prefixes visible from any one switch in the network are reduced exponentially as the flat network is migrated to a hierarchical network.

Figure 11-2 PNNI Hierarchical Network Topology



PNNI hierarchy has certain advantages and disadvantages that you should consider before you decide to implement it in your network.

- An advantage of PNNI hierarchy is its ability to scale to very large networks. This scalability is because of the exponential reduction in size of the visible topology and amount of received topology state information at each switch in the network. These reductions improve the effectiveness of your network by reducing the control traffic, memory, and processing required by each switch in the network.
- A disadvantage of PNNI hierarchy is the loss of information caused by topology aggregation. PNNI performs route computations based on its view of the network topology. Because a hierarchical view of the network is restricted, compared to a nonhierarchical (flat topology) view, routing decisions are not as effective as in a flat topology. In both cases, a path to the destination is selected; however, in most cases the path selected in a flat topology is more efficient. This trade-off between routing efficiency and scalability is not specific to PNNI; it is a known limitation of any hierarchical routing protocol.
- The decision to implement a PNNI hierarchy depends on several factors, including
  - The size of the network

- Type of network traffic
- Call setup activity
- The amount of processing and memory required to handle the PNNI control traffic

Because you must consider several factors, and their interdependency is not easily quantifiable, it is not possible to specify the exact number of nodes above which a flat network must be migrated to a hierarchical network. A high CPU load caused by PNNI control traffic is a strong indication that a hierarchical organization of the topology is required.

ATM Address Description

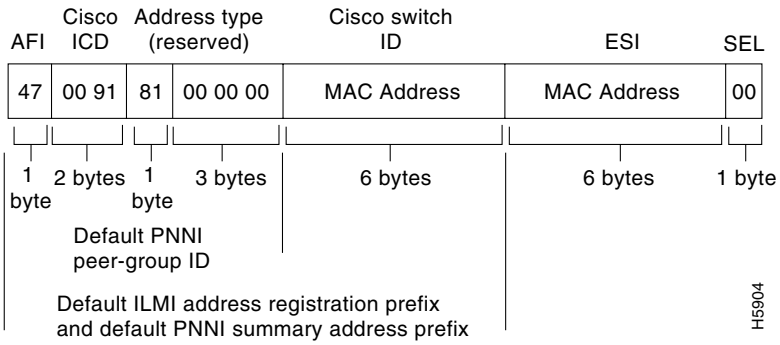
This section describes ATM addresses.

ATM Address Autoconfiguration

The DSLAM is equipped with a preconfigured 20-byte ATM address. This preconfigured address provides plug-and-play operation in isolated flat topology ATM networks. Although the preconfigured addresses are globally unique, they are not suitable for connection to service provider networks or within hierarchical PNNI networks. Furthermore, address summarization is not possible beyond the level of one switch.

The preconfigured ATM address format provided by Cisco Systems is shown in Figure 11-3.

Figure 11-3 Cisco Default ATM Address



All preconfigured addresses share the same 7-byte address prefix. This prefix allows all lowest-level PNNI nodes to generate the same default peer group identifier at level 56. When you interconnect multiple switches, one large autoconfigured peer group is created at level 56. The next six bytes comprise the MAC address of the switch. The 7-byte address prefix combined with the 6-byte MAC address provide a 13-byte prefix that uniquely identifies each switch. This 13-byte prefix is also the default ILMI address prefix and is used by ILMI for address registration and summarization.

ATM Address Formats

The address formats used in PNNI are:

- The ATM End System Address (AESA). AESAs are 20 octets and are derived from the ISO definition of NSAPs. You can further classify AESAs based on the first octet, called the Authority and Format Identifier (AFI). The ATM Forum specifications through UNI Version 4.0 specify only three valid types of AFI: E.164, ICD, and DCC. However, future ATM Forum specifications will allow any AFI that has binary encoding of the Domain Specific Part (DSP) and a length of 20 octets. The DSLAM does not restrict the AFI values.
- E.164 numbers (also known as native E.164 numbers) are supported on UNI and IISP interfaces, but are not directly supported by PNNI. Instead, these are supported indirectly through use of the E.164 AESA format.

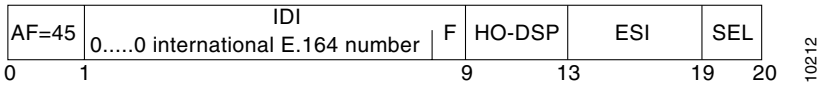
Note See the ATM Forum UNI specifications for more information.

E.164 AESA Prefixes

PNNI address prefixes are usually created by taking the first *p* (0 to 152) bits of an address. Because of the encoding defined for E.164 AESAs, this creates difficulties when native E.164 numbers are used with E.164 AESAs.

The encoding defined for E.164 AESAs in the ATM Forum UNI specifications is shown in Figure 11-4.

Figure 11-4 Normal Encoding of E.164 AESAs (Right-Justified)



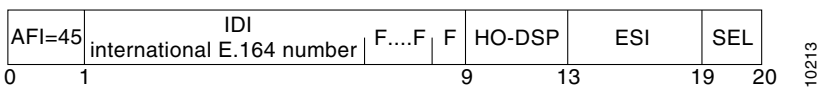
In normal encoding, the international E.164 number is right-justified in the IDI part, with leading semi-octet zeros (0) used to fill any unused spaces. Because the international E.164 number varies in length and is right justified you must configure several E.164 AESA prefixes to represent reachability information to the international E.164 number prefix. These E.164 AESA prefixes differ only in the number of leading zeros between the AFI and the international E.164 number.

For example, all international E.164 numbers that represent destinations in Germany begin with the country code 49. The length of international E.164 numbers in Germany varies between 9 and 12 digits. To configure static routes to all E.164 numbers in Germany, configure static routes to this set of E.164 AESA prefixes:

- 45.00049
- 45.000049
- 45.0000049
- 45.00000049

E.164 numbers that share a common prefix can be summarized by a single reachable address prefix, even when the corresponding set of full E.164 numbers varies in length. For this reason, in PNNI 2.0 the encoding of E.164 address prefixes is modified to a left-justified format, as shown in Figure 11-5.

Figure 11-5 PNNI 2.0 Encoding of E.164 AESAs (Left-Justified)



The left-justified encoding of the international E.164 number within the IDI allows for a single E.164 AESA prefix to represent reachability to all matching E.164 numbers, even when the matching E.164 numbers vary in length. Before PNNI routing looks up a destination address to find a route to that address, it converts the destination address from the call setup in the same way and then carries out the longest match lookup.

Note The converted encoding of the E.164 AESA is not used in PNNI signaling, even in PNNI 2.0. The conversion is only used for PNNI reachable address prefixes, and when determining the longest matching address prefix for a given AESA. Full 20-byte AESAs are always encoded as shown in Figure 11-4.

The DSLAM supports the PNNI 2.0 encoding of E.164 AESAs with the **aesa embedded-number left-justified** command. When you enter this command, all reachable address prefixes with the E.164 AFI are automatically converted into the left-justified encoding format. This includes reachable address prefixes advertised by remote PNNI nodes, ATM static routes, summary address prefixes, routes learned by ILMI, and reachable address prefixes installed by the switch automatically (that is, representing the switch address and the soft PVC addresses on this switch). This affects the **atm route, auto-summary**, **summary-address**, **show atm route**, and **show atm pnni summary** commands. The **atm address**, **atm prefix**, and **show atm addresses** commands are not affected because they do not use PNNI address prefixes.

Note All switches or ATM DSLAMs in the PNNI routing domain must have the same configuration by entering the **aesa embedded-number left-justified** command.

Obtaining ATM Addresses

You can categorize ATM addresses by ownership: customer-owned ATM addresses and service provider ATM addresses.

If you have a private network, you can obtain ATM addresses from these sources:

- An ATM service provider—Any AESA format is acceptable.
- The national registration authority— In the United States, the national registration authority is ANSI. In the United Kingdom, the national registration authority is FEI.

In customer-owned ATM addresses, the main part of the address is allocated directly to a private networking customer by a national or world registration authority. A customer owned ATM address (owned by Cisco) is preconfigured on each DSLAM. If you do not implement a hierarchy in your PNNI network, you can use the preconfigured ATM address.

In service provider ATM addresses, the main part of the address is allocated to the network operator by the appropriate national or world registration authority. The operator may then suballocate part of the address space to customers.

ATM service providers can obtain these types of ATM addresses:

- E.164 numbers or E.164 AESAs from the ITU or the national numbering authority.
- ICD AESAs from the British Standards Institute (BSI) by way of a national registration authority.
- DCC AESAs from the national registration authority. In the U.S.A., the national registration authority is American National Standards Institute (ANSI). In the United Kingdom, the national registration authority is FEI.

Designing an ATM Address Plan

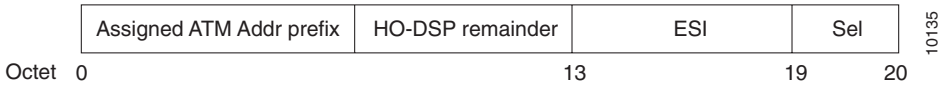
Your ATM address plan is key to efficient operation and management of PNNI networks. When you design an ATM address plan, the most important points to remember are:

- Your ATM address prefixes must be globally unique.
- The addresses must be hierarchical, corresponding to your network topology.
- You must plan for future network expansion.

Globally Unique ATM Address Prefixes

You can obtain globally unique address prefixes from a national or world registration authority or they can be suballocated to you from a service provider's address space. Make sure that the addresses you assign in your network are derived from a globally unique address prefix, as shown in Figure 11-6.

Figure 11-6 Unique ATM Address Prefix Used to Assign ATM Addresses



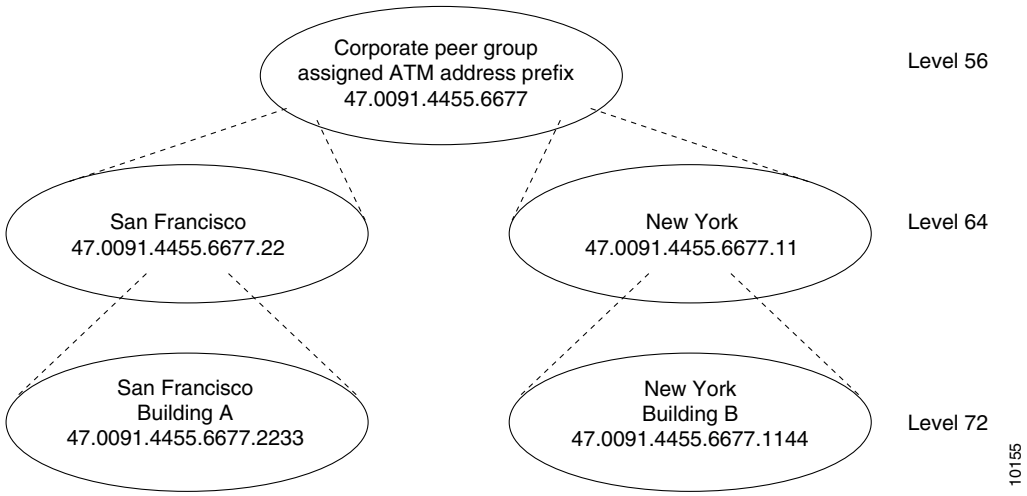
For more information, refer to the "Obtaining ATM Addresses" section on page 11-198.

Hierarchical Addresses

The HO-DSP remainder, the part of the address between the assigned ATM address prefix and the ESI, should be assigned in a hierarchical manner. All systems in the network share the assigned ATM address prefix.

You can further subdivide the assigned address space by providing longer prefixes to different regions of the network. Within each peer group, be certain that the first level bits of each switch address matches the corresponding bits of the Peer Group Identifier (PGI) value. An example of a hierarchical address assignment is shown in Figure 11-7.

Figure 11-7 Sample Hierarchical Address Assignment



Note that the address prefix is longer at each lower level of the PNNI hierarchy shown in Figure 11-7.

The advantages of hierarchical address assignment include

- Greatly increased scalability by minimizing the number of PNNI routes stored and processed by each node
- Simplified configuration and management of the PNNI hierarchy

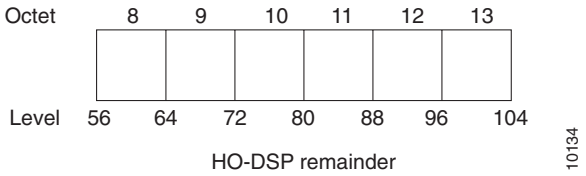
When the ATM network topology (which consists of switches, links, and virtual path [VP] tunnels) differs from the logical topology (which consists of VPNs and virtual LANs), it is important that the address hierarchy follow the network topology. You can construct the logical topology using other features, such as emulated LANs or Closed User Groups (CUGs).

Planning for Future Growth

When you are constructing the address hierarchy, it is important to plan ahead for the maximum number of levels that you might need for future growth. Not all levels in the addressing hierarchy need to be used by PNNI. It is possible to run with fewer PNNI levels in the beginning, and then migrate to more levels of hierarchy in the future. For example, you can configure the network as one large peer group where the PGI value is based on the assigned ATM address prefix. By planning ahead, you can easily migrate to more levels of hierarchy without manually renumbering all of the switches and end systems.

You can subdivide the HO-DSP remainder to allow for upward and downward future growth. For example, assume that you have 6 octets available for the HO-DSP remainder: 8 through 13 (as shown in Figure 11-8).

Figure 11-8 HO-DSP Remainder Subdivision Example



The HO-DSP remainder in this example spans levels 56 through 104. To allow for future expansion at the lowest level of the hierarchy, you must provide sufficient addressing space in the HO-DSP remainder to accommodate all future switches.

Assume that you start with the lowest level at 88. For administrative purposes, in the future you might want to group some of these switches into peer groups where additional switches will be added. For those switches that will be part of the new peer group you should assign addresses that can be easily clustered into a level 96 peer group. These addresses share a common 12th octet, leaving the 13th octet for downward future expansion.

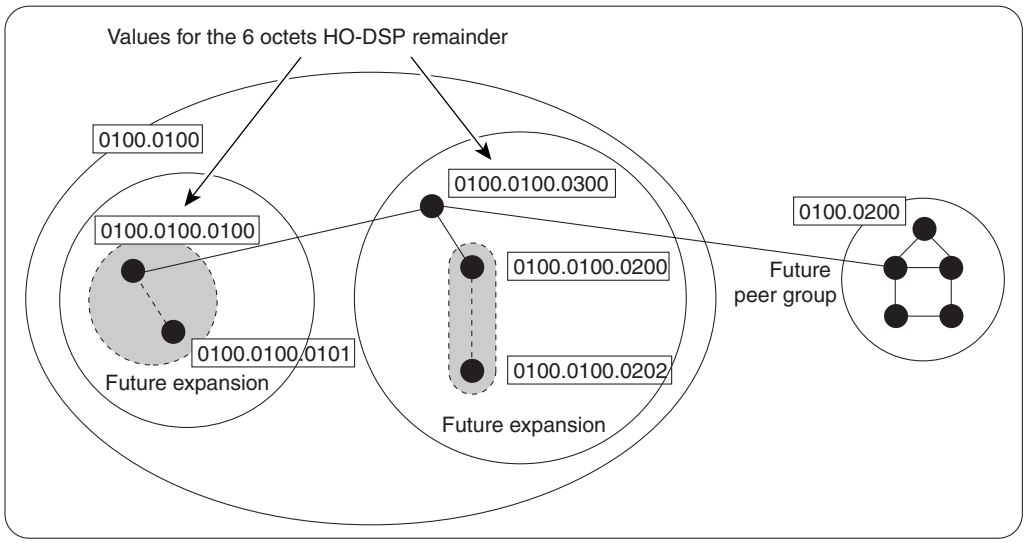
The octet pairs (12 and 13) for these switches could be: (01, 00), (02, 00), (03, 00) and so on, while switches that will be added in the future could be: (02, 01), (02, 02), (02, 03) and so on.

This type of addressing scheme leaves room for expansion without requiring address modification. If you add a hierarchical level 96, the switches will form a new peer group at level 96.

Although you started with no more than 256 switches at the lowest level, by expanding this to two levels in the future, you are able to accommodate up to 65,536 switches in the same region.

Figure 11-9 shows an example of HO-DSP assignment.

Figure 11-9 Example of HO-DSP Assignment for Future Expansion



By following similar guidelines, you can plan for future expansion in the upward and downward direction. Specifically, you can expand upward by adding hierarchical levels as your network grows in size.

Configuring IISP

This section describes the procedures necessary for IISP configuration.

Configuring the Routing Mode

You can restrict the ATM routing software to operate in static mode. In this mode, call routing is restricted to only the static configuration of ATM routes, disabling operation of any dynamic ATM routing protocols, such as PNNI.

The **atm routing-mode** command is different from deleting all PNNI nodes using the **node** command and affects ILMI autoconfiguration. If the switch or DSLAM is configured using static routing mode on each interface, the switch ILMI variable atmfAtmLayerNniSigVersion is set to IISP. This causes either of these events to occur:

- ILMI autoconfiguration on the interfaces between two switches determines the interface type as IISP.
- The switch on the other side indicates that the Network-to-Network Interface (NNI) signaling protocol is not supported.

Note The **atm routing-mode** command is activated only after the next software reload. The switch continues to operate in the current mode until the software is reloaded.

To configure the routing mode, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm routing-mode static**	Configure the ATM routing mode to static.
2	**end**	Exit configuration mode.
3	**copy running-config startup-config**	Write the running configuration to the startup configuration.
4	**reload**	Reload the switch software.

Example

This example shows how to use the **atm routing-mode static** command to restrict the switch operation to static routing mode and displays the result:

```
DSLAM(config)# atm routing-mode static
This Configuration Will Not Take Effect Until Next Reload.

DSLAM(config)# end

DSLAM# copy running-config startup-config
Building configuration...
[OK]

DSLAM# reload

DSLAM# show running-config
Building configuration...

Current configuration:
!
version 11.2
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
!
username dtate
ip rcmd remote-username dplatz
!
atm e164 translation-table
 e164 address 1111111 nsap-address 11.111111111111111111111111.112233445566.11
 e164 address 2222222 nsap-address 22.222222222222222222222222.112233445566.22
 e164 address 3333333 nsap-address 33.333333333333333333333333.112233445566.33
!
atm routing-mode static
atm address 47.0091.8100.0000.0040.0b0a.2b81.0040.0b0a.2b81.00
!
interface CBR0/0
 no ip address

<Information Deleted>
```

This example shows how to reset the switch operation back to PNNI if the DSLAM is operating in static mode:

```
DSLAM(config)# no atm routing-mode static
This Configuration Will Not Take Effect Until Next Reload.
```

```
DSLAM(config)# end

DSLAM# copy running-config startup-config
Building configuration...
[OK]

DSLAM# reload
```

Configuring the ATM Address

If you are planning to implement only a flat topology network (and have no future plans to migrate to PNNI hierarchy), you can skip this section and use the preconfigured ATM address assigned by Cisco Systems.

Note For information about ATM address considerations, refer to the "ATM Address Description" section on page 11-196.

To change the active ATM address follow these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm address** *atm-address*	Configure the ATM address for the DSLAM.
2	**end**	Return to privileged EXEC mode.
3	**show atm addresses**	Verify the new address.
4	**configure** [**terminal**]	Enter configuration mode from the terminal.
5	**no atm address** *atm_address*	At the configuration mode prompt, remove the old ATM address from the DSLAM.

Example

This example shows how to add the ATM address prefix 47.0091.8100.5670.000.0ca7.ce01 and remove the old address from the DSLAM and displays the result. Using the ellipses (...) adds the default Media Access Control (MAC) address as the last six bytes.

```
DSLAM(config)# atm address 47.0091.8100.5670.0000.0ca7.ce01...

DSLAM(config)# no atm address 47.0091.8100.0000.0041.0b0a.1081...

DSLAM# show atm addresses

Switch Address(es):
  47.00918100000000410B0A1081.00410B0A1081.00 active
  47.0091810056700000CA7CE01.00410B0A1081.00

Soft VC Address(es):
Soft VC Address(es):
  47.0091.8100.0000.007b.f444.7801.4000.0c80.0010.00 ATM0/1
  47.0091.8100.0000.007b.f444.7801.4000.0c80.0020.00 ATM0/2

ILMI Switch Prefix(es):
```

```
  47.0091.8100.0000.007b.f444.7801

ILMI Configured Interface Prefix(es):

LECS Address(es):
```

Configuring Static Routes

Use the **atm route** command to configure a static route. A static route attached to an interface allows all ATM addresses matching the configured address prefix to be reached through that interface.

Note For private UNIs where ILMI address registration is not used, internal-type static routes should be configured to a 19-byte address prefix representing the attached end system.

To configure a static route, use this global configuration command:

Command	Task
atm route *atm-address-prefix* **atm** *slot/port* [**e164-address** *e164-address* [**number-type** {**international** | **local** | **national** | **subscriber**}] [**internal**] [**scope** *1-15*]	Specify a static route to a reachable address prefix.

Examples

This example uses the **atm route** command to configure a static route to the 13-byte switch prefix 47.00918100000000410B0A1081 to ATM interface 0/0:

```
DSLAM(config)# atm route 47.00918100000000410B0A1081 atm 0/0
```

This example uses the **atm route** command to configure a static route to the 13-byte switch prefix 47.00918100000000410B0A1081 to ATM interface 0/0 configured with a scope 1 associated:

```
DSLAM(config)# atm route 47.0091.8100.0000 atm 0/0 scope 1
```

This example shows the ATM static route configuration using the **show atm route** EXEC command:

```
DSLAM# show atm route

Codes: P - installing Protocol (S - Static, P - PNNI, R - Routing control),
       T - Type (I - Internal prefix, E - Exterior prefix, SE -
                 Summary Exterior prefix, SI - Summary Internal prefix,
                 ZE - Suppress Summary Exterior, ZI - Suppress Summary Internal)

P  T Node/Port        St Lev Prefix
~ ~~ ~~~~~~~~~~~~~~~~ ~~ ~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
S  E 1   ATM0/0     DN 56  47.0091.8100.0000/56
S  E 1   ATM0/0     DN 0   47.0091.8100.0000.00/64
                            (E164 Address 1234567)
R SI 1   0         UP 0   47.0091.8100.0000.0041.0b0a.1081/104
R  I 1   ATM0/0    UP 0   47.0091.8100.0000.0041.0b0a.1081.0041.0b0a.1081/152
R  I 1   ATM0/0    UP 0   47.0091.8100.0000.0041.0b0a.1081.4000.0c/128
R SI 1   0             UP 0   47.0091.8100.5670.0000.0000.0000/104
R  I 1     ATM0/0     UP 0   47.0091.8100.5670.0000.0000.0000.0040.0b0a.1081/152
R  I 1   ATM0/0     UP 0   47.0091.8100.5670.0000.0000.0000.4000.0c/128
```

Configuring PNNI

This section describes all of the procedures necessary for you to create a basic PNNI configuration.

Configuring PNNI Without Hierarchy

The DSLAM defaults to a working PNNI configuration suitable for operation in isolated flat topology ATM networks. The DSLAM comes with a globally unique preconfigured ATM address. Manual configuration is not required if you

- Have a flat network topology
- Do not plan to connect the DSLAM to a service provider network
- Do not plan to migrate to a PNNI hierarchy in the future

If you plan to migrate your flat network topology to a PNNI hierarchical topology, proceed to the next section.

Configuring the Lowest Level of the PNNI Hierarchy

This section describes how to configure the lowest level of the PNNI hierarchy. The lowest-level nodes comprise the lowest level of the PNNI hierarchy. When only the lowest-level nodes are configured, there is no hierarchical structure. If your network is relatively small and you want the benefits of PNNI, but do not need the benefits of a hierarchical structure, follow the procedures in this section to configure the lowest level of the PNNI hierarchy.

To implement multiple levels of PNNI hierarchy, first complete the procedures in this section and then proceed to the "Configuring Higher Levels of the PNNI Hierarchy" section on page 11-211.

The lowest level PNNI configuration includes these procedures:

- Configuring an ATM Address and PNNI Node Level
- Configuring Static Routes
- Configuring a Summary Address
- Configuring Scope Mapping

Configuring an ATM Address and PNNI Node Level

If you are planning to implement a:

- Flat topology network (and have no future plans to migrate to PNNI hierarchy), you can skip this section and use the preconfigured ATM address assigned by Cisco Systems.
- PNNI hierarchy, follow the procedure in this section to configure an ATM address and the PNNI node level.

The DSLAM is preconfigured as a single lowest-level PNNI node (locally identified as node 1) with a level of 56. The system calculates the node ID and peer group ID based on the current active ATM address.

To configure a node in a higher level of the PNNI hierarchy, the value of the node level must be a smaller number than the previous node. For example, a three-level hierarchical network could progress from level 72 to level 64 to level 56. Notice that the level numbers graduate from largest at the lowest level (72) to smallest at the highest level (56). (See Figure 11-7 earlier in this chapter.)

To change the active ATM address, create a new address, verify that it exists, and then delete the current active address. After you have entered the new ATM address, disable node 1 and then reenable it. At the same time, you can change the node level if required for your configuration. The identifiers for all higher level nodes are recalculated based on the new ATM address.

Caution The system does not recalculate node IDs and peer group IDs until the node is disabled and then re-enabled.

Note For information about ATM address considerations, refer to the "ATM Address Description" section on page 11-196.

To change the active ATM address, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm address** *atm_address*	At the configuration mode prompt, configure the new ATM address for the DSLAM.
2	**end**	Return to privileged EXEC mode.
3	**show atm addresses**	Verify the new address.
4	**configure** [**terminal**]	Enter configuration mode from the terminal.
5	**no atm address** *atm_address*	At the configuration mode prompt, remove the old ATM address from the DSLAM.
6	**atm router pnni**	At the configuration mode prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
7	**node 1 disable**	At the configure ATM router prompt, disable the PNNI node.
8	**node 1 level** *level* **enable**	Reenable the node. You can also change the node level if required for your configuration.

Example

This example changes the ATM address of the DSLAM from the autoconfigured address 47.0091.8100.0000.0041.0b0a.1081.0041.0b0a.1081.00 to the new address prefix 47.0091.8100.5670.0000.0000.1122.0041.0b0a.1081.00, and causes the node identifier and peer group identifier to be recalculated:

```
DSLAM(config)# atm address 47.0091.8100.5670.0000.0000.1122...

DSLAM(config)# no atm address 47.0091.8100.0000.0041.0b0a.1081...
```

```
DSLAM(config)# atm router pnni

DSLAM(config-atm-router)# node 1 disable

DSLAM(config-pnni-node)# node 1 enable
```

This example shows the PNNI node configuration using the show **atm pnni local-node** privileged EXEC command:

```
DSLAM# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: eng_1
  System address          47.009181000000002EB1FFE00.0002EB1FFE00.01
  Node ID          56:160:47.009181000000002EB1FFE00.0002EB1FFE00.00
  Peer group ID        56:160:47.0000.0000.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 1, No. of neighbors 0
  Parent Node Index: 2
  Node Allows Transit Calls
  Node Representation: simple

  Hello interval 15 sec, inactivity factor 5,
  Hello hold-down 10 tenths of sec
  Ack-delay 10 tenths of sec, retransmit interval 5 sec,
  Resource poll interval 5 sec
  SVCC integrity times: calling 35 sec, called 50 sec,
  Horizontal Link inactivity time 120 sec,
  PTSE refresh interval 1800 sec, lifetime factor 200 percent,
  Min PTSE interval 10 tenths of sec
  Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
  Default administrative weight mode: uniform
  Max admin weight percentage: -1
  Next resource poll in 3 seconds
  Max PTSEs requested per PTSE request packet: 32
  Redistributing static routes: Yes
```

Configuring Static Routes

Because PNNI is a dynamic routing protocol, static routes are not required between nodes that support PNNI. However, you can extend the routing capability of PNNI beyond nodes that support PNNI to

- Connect to nodes outside of a peer group that do not support PNNI
- Define routes to end systems that do not support ILMI

Use the **atm route** command to configure a static route. A static route attached to an interface allows all ATM addresses matching the configured address prefix to be reached through that interface.

Note You can connect two PNNI peer groups using the IISP protocol. Connecting PNNI peer groups requires that you configure a static route on the IISP interfaces, allowing connections to be set up across the IISP links.

To configure a static route connection, use this global configuration command:

Command	Task
atm route *atm-address-prefix* **atm** *slot/port* [**e164-address** *e164-address* [**number-type** {**international** | **local** | **national** | **subscriber**}]] [**internal**] [**scope** *1-15*]	Specify a static route to a reachable address prefix.

Examples

This example uses the **atm route** command to configure a static route to the 13-byte switch prefix 47.00918100000000410B0A1081 to ATM interface 0/0:

```
DSLAM(config)# atm route 47.00918100000000410B0A1081 atm 0/0
```

This example uses the **atm route** command to configure a static route to the 13-byte switch prefix 47.00918100000000410B0A1081 to ATM interface 0/0 configured with a scope 1 associated:

```
DSLAM(config)# atm route 47.0091.8100.0000 atm 0/0 scope 1
```

This example shows the ATM static route configuration using the **show atm route** EXEC command:

```
DSLAM# show atm route

Codes: P - installing Protocol (S - Static, P - PNNI, R - Routing control),
       T - Type (I - Internal prefix, E - Exterior prefix, SE -
                 Summary Exterior prefix, SI - Summary Internal prefix,
                 ZE - Suppress Summary Exterior, ZI - Suppress Summary Internal)

P  T Node/Port        St Lev Prefix
~ ~~ ~~~~~~~~~~~~~~~~ ~~ ~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
S  E 1   ATM0/0     DN 56  47.0091.8100.0000/56
S  E 1   ATM0/0     DN 0   47.0091.8100.0000.00/64
                             (E164 Address 1234567)
R SI 1   0            UP 0   47.0091.8100.0000.0041.0b0a.1081/104
R  I 1     ATM0/0     UP 0   47.0091.8100.0000.0041.0b0a.1081.0041.0b0a.1081/152
R  I 1   ATM0/0     UP 0   47.0091.8100.0000.0041.0b0a.1081.4000.0c/128
R SI 1   0            UP 0   47.0091.8100.5670.0000.0000.0000/104
R  I 1     ATM0/0     UP 0   47.0091.8100.5670.0000.0000.0000.0040.0b0a.1081/152
R  I 1   ATM0/0     UP 0   47.0091.8100.5670.0000.0000.0000.4000.0c/128
```

Configuring a Summary Address

You can configure summary addresses to reduce the amount of information advertised by a PNNI node and contribute to scalability in large networks. Each summary address consists of a single reachable address prefix that represents a collection of end system or node addresses.

We recommend that you use summary addresses when all end system addresses that match the summary address are directly reachable from the node. However, this is not always required because routes are always selected by nodes advertising the longest matching prefix to a destination address.

By default, each lowest-level node has a summary address equal to the 13-byte address prefix of the ATM address of the DSLAM. This address prefix is advertised into its peer group.

You can configure multiple addresses for a single DSLAM which are used during ATM address migration. ILMI registers end systems with multiple prefixes during this period until an old address is removed. PNNI automatically creates 13-byte summary address prefixes from all of its ATM addresses.

You must configure summary addresses (other than the defaults) on each node. Each node can have multiple summary address prefixes. Use the **summary-address** command to manually configure summary address prefixes.

Note The **no auto-summary** command removes the default summary addresses. Use the **no auto-summary** command when systems that match the first 13 bytes of the ATM addresses of your DSLAM are attached to different devices. You can also use this command for security purposes.

To configure a summary address, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**no auto-summary**	Remove the default summary addresses.
4	**summary-address** *address_prefix*	Configure the ATM PNNI summary address prefix.

Examples

This example removes the default summary addresses and adds summary address 47.009181005670:

```
DSLAM(config)# atm router pnni

DSLAM(config-atm-router)# node 1

DSLAM(config-pnni-node)# no auto-summary

DSLAM(config-pnni-node)# summary-address 47.009181005670
```

This example shows the ATM PNNI summary address configuration using the **show atm pnni summary** privileged EXEC command:

```
DSLAM# show atm pnni summary

Codes: Node - Node index advertising this summary
       Type - Summary type (INT - internal, EXT - exterior)
       Sup  - Suppressed flag (Y - Yes, N - No)
       Auto - Auto Summary flag (Y - Yes, N - No)
       Adv  - Advertised flag (Y - Yes, N - No)

 Node Type Sup Auto Adv  Summary Prefix
 ~~~~ ~~~~ ~~~ ~~~~ ~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  1   Int   N   Y    Y   47.0091.8100.0000.0040.0b0a.2a81/104
  2   Int   N   Y    N   47.01b1.0000.0000.0000.00/80
```

Configuring Scope Mapping

The PNNI address scope allows you to restrict advertised reachability information within configurable boundaries.

Note On UNI and IISP interfaces, the scope is specified in terms of organizational scope values ranging from 1 (local) to 15 (global). (Refer to the ATM Forum UNI Signaling Version 4.0 specification for more information.)

In PNNI networks, the scope is specified in terms of PNNI levels. The mapping from organizational scope values used at UNI and IISP interfaces to PNNI levels is configured on the lowest-level node. The mapping can be determined automatically (which is the default setting) or manually, depending on the configuration of the **scope mode** command.

In manual mode, if you modify the level of node 1, make sure you also reconfigure the scope map to avoid unintended suppression of reachability advertisements. Misconfiguring the scope map could cause addresses to remain unadvertised.

In automatic mode, the UNI to PNNI level mapping is automatically reconfigured each time the level of the node 1 is modified. The automatic reconfiguration prevents misconfigurations caused by node level modifications. Automatic adjustment of scope mapping uses the values shown in Table 11-1.

Table 11-1 Scope Mapping Table

Organizational Scope	ATM Forum PNNI 1.0 Default Level	Automatic Mode PNNI Level
1 to 3	96	Minimum (l,96)
4 to 5	80	Minimum (l,80)
6 to 7	72	Minimum (l,72)
8 to 10	64	Minimum (l,64)
11 to 12	48	Minimum (l,48)
13 to 14	32	Minimum (l,32)
15 (global)	0	0

If you enter the **scope mode automatic** command, this ensures that all organizational scope values cover an area at least as wide as the current node's peer group. Configuring the scope mode to **manual** disables this feature and no changes can be made without explicit configuration.

To configure the PNNI scope mapping, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**scope mode** {**automatic** | **manual**}	Configure scope mode as manual.[1]
4	**scope map** *low-org-scope* [*high-org-scope*] **level** *level-number*	Configure node scope mapping.

1. You must enter the **scope mode manual** command to allow scope mapping configuration.

Example

This example shows how to configure PNNI scope mapping manually so that organizational scope values 1 through 8 map to PNNI level 72 and displays the result:

```
DSLAM(config)# atm router pnni

DSLAM(config-atm-router)# node 1

DSLAM(config-pnni-node)# scope mode manual

DSLAM(config-pnni-node)# scope map 1 8 level 72

DSLAM# show atm pnni scope
UNI scope   PNNI Level
~~~~~~~~~   ~~~~~~~~~~
(1  - 10)      56
(11 - 12)      48
(13 - 14)      32
(15 - 15)      0

Scope mode: manual
```

Configuring Higher Levels of the PNNI Hierarchy

This section describes the procedures to configure higher levels of PNNI hierarchy.

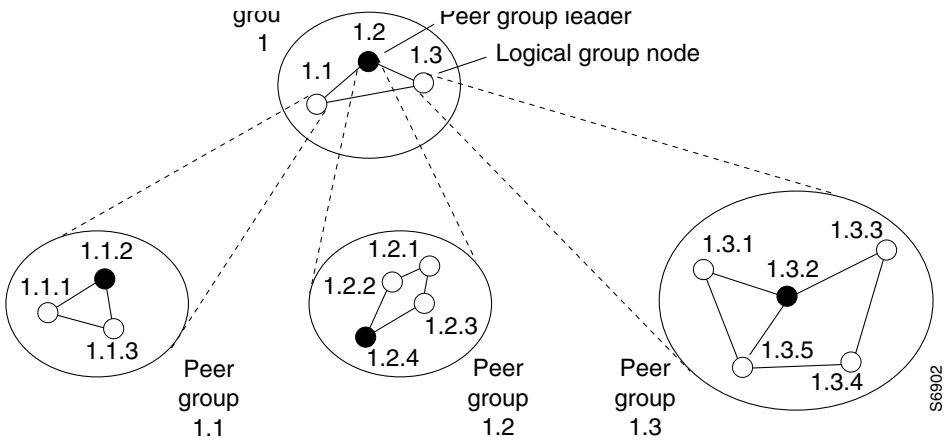
After you have configured the lowest level of the PNNI hierarchy (see the section, "Configuring the Lowest Level of the PNNI Hierarchy" section on page 11-205), you can complete the PNNI hierarchical structure by configuring peer group leaders (PGLs) and logical group nodes (LGNs).

Each peer group can contain one active PGL. The PGL is a logical node within the peer group that collects data about the peer group to represent it as a single node to the next PNNI hierarchical level.

Upon becoming a PGL, the PGL creates a parent LGN. The LGN represents the PGL's peer group within the next higher level peer group. The LGN aggregates and summarizes information about its child peer group and floods that information into its own peer group.

The LGN also distributes information received from its peer group to the PGL of its child peer group for flooding. Figure 11-10 shows an example of PGLs and LGNs.

Figure 11-10 PGLs and LGNs



To create the PNNI hierarchy, select DSLAMs that are eligible to become PGLs at each level of the hierarchy. Nodes can become PGLs through the peer group leader election process. Each node has a configured election priority.

To be eligible for election, the configured priority must be greater than zero and a parent node must be configured. Normally the node with the highest configured leadership priority in a peer group is elected PGL. You can configure multiple nodes in a peer group with a non-zero leadership priority so that if one PGL becomes unreachable, the node configured with the next highest election leadership priority becomes the new PGL.

Note The choice of PGL does not directly affect the selection of routes across a peer group.

Because any one peer group can consist of both lowest level nodes and LGNs, lowest level nodes should be preferred as PGLs. Configuring the network hierarchy with multiple LGNs at the same DSLAM creates additional PNNI processing and results in slower recovery from failures. Selecting DSLAMs for election with more processing capability (for example, because a smaller volume of call processing compared to others) may be better.

We recommend that each node in a peer group that can become a PGL be assigned the same parent node configuration.

Configuring a Logical Group Node and Peer Group Identifier

You can configure a new LGN by entering the **node** command with an unused node index value between 2 and 8.

The LGN is created only when the child node in the same DSLAM (that is, the node whose parent configuration points to this node) is elected PGL of the child peer group.

The peer group identifier defaults to a value created from the first part of the child peer group identifier, and does not need to be specified. If you want a non-default peer group identifier, you must configure all logical nodes within a peer group with the same peer group identifier.

Higher level nodes only become active if

- A lower-level node specifies the higher-level node as a parent.
- The election leadership priority of the child node is configured with a non-zero value and is elected as the PGL.

To configure a LGN and peer group identifier, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	Enter ATM router PNNI mode. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index* **level** *level* [**lowest**] [**peer-group-identifier** *dd:xxx*] [**enable** | **disable**]	Configure the logical node and optionally its peer group identifier. Configure each logical node in the peer group with the same peer group identifier. When you have more than one logical node on the same DSLAM, you must specify a different index number to distinguish it from node 1.

Example

This example shows how to create a new node 2 with a level of 56 and a peer group identifier of 56:47009111223344 and displays the result. Notice that the PNNI level and the first two digits of the peer group identifier are the same:

```
DSLAM(config)# atm router pnni

DSLAM(config-atm-router)# node 2 level 56 peer-group-identifier 56:47009111223344 enable

DSLAM(config-pnni-node)# end

DSLAM# show atm pnni local-node 2

PNNI node 2 is enabled and not running
  Node name: Switch.2.56
  System address          47.009181000000000000000001.000000000001.02
  Node ID            56:0:00.000000000000000000000000.000000000001.00
  Peer group ID        56:47.0091.1122.3344.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: NONE
  Node Allows Transit Calls
  Node Representation: simple

  Hello interval 15 sec, inactivity factor 5,
  Hello hold-down 10 tenths of sec
  Ack-delay 10 tenths of sec, retransmit interval 5 sec,
  Resource poll interval 5 sec
  SVCC integrity times: calling 35 sec, called 50 sec,
  Horizontal Link inactivity time 120 sec,
  PTSE refresh interval 1800 sec, lifetime factor 200 percent,
  Min PTSE interval 10 tenths of sec
  Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
  Default administrative weight mode: uniform
  Max admin weight percentage: -1
  Max PTSEs requested per PTSE request packet: 32
  Redistributing static routes: No
```

Configuring the Node Name

PNNI node names default to names based on the host name. For example, if the host name is SanFran1, the default node name is also SanFran1. If you prefer node names that more accurately reflect the peer group, you can use the **name** command to change the default node name. For example, you could change the node name to Cal1 to represent the entire location of the peer group to which it belongs. Cisco recommends you choose a node name of 12 characters or less so that your screen displays remain well formatted and easy to read.

After you configure a node name, the system distributes it to all other nodes by PNNI flooding. This allows the node to be identified by its node name in PNNI **show** commands.

To configure the PNNI node name, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.

Step	Command	Task
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**name** *name_string*	Configure the node name.

Example

This example configures the name of the node as **eng_1** using the **name** command, and displays the result:

```
DSLAM(config)# atm router pnni

DSLAM(config-atm-router)# node 1

DSLAM(config-pnni-node)# name eng_1

DSLAM# show atm pnni local-node
PNNI node 1 is enabled and running
  Node name: eng_1
  System address          47.0091810000000002EB1FFE00.0002EB1FFE00.01
  Node ID          56:160:47.0091810000000002EB1FFE00.0002EB1FFE00.00
  Peer group ID        56:16.0347.0000.0000.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 1, No. of neighbors 0
  Parent Node Index: 2
  Node Allows Transit Calls
  Node Representation: simple

  Hello interval 15 sec, inactivity factor 5,
  Hello hold-down 10 tenths of sec
  Ack-delay 10 tenths of sec, retransmit interval 5 sec,
  Resource poll interval 5 sec
  SVCC integrity times: calling 35 sec, called 50 sec,
  Horizontal Link inactivity time 120 sec,
  PTSE refresh interval 1800 sec, lifetime factor 200 percent,
  Min PTSE interval 10 tenths of sec
  Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
  Default administrative weight mode: uniform
  Max admin weight percentage: -1
  Next resource poll in 3 seconds
  Max PTSEs requested per PTSE request packet: 32
  Redistributing static routes: Yes
```

Configuring a Parent Node

For a node to be eligible to become a PGL within its own peer group, you must configure a parent node and an election leadership level (described in the section "Configuring the Node Election Leadership Priority" section on page 11-215). If the node is elected a PGL, the node specified by the **parent** command becomes the parent node and represents the peer group at the next hierarchical level.

To configure a parent node, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**parent** *node_index*	Configure the parent node index.

Example

This example creates a parent node for node 1 and displays the result:

```
DSLAM(config)# atm router pnni

DSLAM(config-pnni-node)# node 1

DSLAM(config-pnni-node)# parent 2

DSLAM# show atm pnni hierarchy
Locally configured parent nodes:
  Node          Parent
  Index  Level  Index   Local-node Status     Node Name
  ~~~~~  ~~~~~  ~~~~~~  ~~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~~~~
  1      80     2       Enabled/ Running      DSLAM
  2      72     N/A     Enabled/ Running      DSLAM.2.72
```

Configuring the Node Election Leadership Priority

Normally the node with the highest election leadership priority is elected PGL. If two nodes share the same election priority, the node with the highest node identifier becomes the PGL. To be eligible for election, ensure that the configured priority is greater than zero. You can configure multiple nodes in a peer group with non-zero leadership priority so that if one PGL becomes unreachable, the node configured with the next highest election leadership priority becomes the new PGL.

Note The choice of PGL does not directly affect the selection of routes across the peer group.

The control for election is done through the assignment of leadership priorities. We recommend that the leadership priority space be divided into three tiers:

- First tier—1 to 49
- Second tier—100 to 149
- Third tier—200 to 205

This subdivision exists because of the GroupLeaderIncrement variable. When a node becomes PGL, it increases the advertised leadership priority by a value of 50 to avoid instabilities after election.

Keep nodes that you do not want to become PGLs assigned to a default leadership priority value of 0.

If among the PGL candidates no node must be forced to be PGL, then assign all leadership priority values within the first tier. After a node is elected PGL, it remains PGL until it steps down a tier or is configured to step down.

If certain nodes must take precedence over nodes in the first tier, even if one is already PGL, leadership priority values can be assigned from the second tier. We recommend that you configure more than one node with a leadership priority value from the second tier. This prevents one unstable node with a larger leadership priority value from destabilizing the peer group repeatedly.

If you need a strict master leader, use the third tier.

Note The **election leadership-priority** command does not take effect unless you configured a parent node using the **node** and **parent** commands.

To configure the election leadership priority, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	Enter ATM router PNNI mode from the terminal.
2	**node** *node_index*	Enter node configuration mode.
3	**election leadership-priority** *number*	Configure the election leadership priority. The configurable range is from 0 to 205.

Example

This example changes the election leadership priority for node 1 to 100 and displays the result:

```
DSLAM(config)# atm router pnni

DSLAM(config-pnni-node)# node 1

DSLAM(config-pnni-node)# election leadership-priority 100

DSLAM# show atm pnni election

PGL Status.............: PGL
Preferred PGL..........: (1) Switch
Preferred PGL Priority.: 255
Active PGL.............: (1) Switch
Active PGL Priority....: 255
Active PGL For.........: 00:01:07
Current FSM State......: PGLE Operating: PGL
Last FSM State.........: PGLE Awaiting Unanimity
Last FSM Event.........: Unanimous Vote

Configured Priority....: 205
Advertised Priority....: 255
Conf. Parent Node Index: 2
PGL Init Interval......: 15 secs
Search Peer Interval...: 75 secs
Re-election Interval...: 15 secs
Override Delay.........: 30 secs
```

This example shows all nodes in the peer group using the **show atm pnni election peers** command:

```
DSLAM# show atm pnni election peers

  Node No.   Priority   Connected   Preferred PGL
  ~~~~~~~~   ~~~~~~~~   ~~~~~~~~~   ~~~~~~~~~~~~~
```

```
1          255        Yes         Switch
9          0          Yes         Switch
10         0          Yes         Switch
11         0          Yes         Switch
12         0          Yes         Switch
```

Configuring a Summary Address

You can use summary addresses to decrease the amount of information advertised by a PNNI node, and thereby contribute to scaling in large networks. Each summary address consists of a single reachable address prefix that represents a collection of end system or node addresses that begin with the given prefix. Only use summary addresses when all end system addresses that match the summary address are directly reachable from this node. However, this is not always required because routes are always selected to nodes advertising the longest matching prefix to a destination address.

Configure a single default summary address for each logical group node (LGN) in the PNNI hierarchy. The length of that summary for any LGN equals the level of the child peer group, and its value is equal to the first level bits of the child peer group identifier. This address prefix is advertised into the LGN's peer group.

Explicitly configure summary addresses other than defaults on each node. Use the **summary-address** command to manually configure summary address prefixes. A node can have multiple summary address prefixes.

Assign the same summary address lists to each node in a peer group that has a potential to become a PGL for its parent node configuration.

Note The **no auto-summary** command removes the default summary addresses. Use the **no auto-summary** command when systems that match the first 13 bytes of the ATM addresses of your DSLAM are attached to different DSLAMs.

To configure the ATM PNNI summary address prefix, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**no auto-summary**	Remove the default summary addresses.
4	**summary-address** *address_prefix*	Configure the ATM PNNI summary address prefix.

Example

This example shows how to remove the default summary addresses and add summary address 47.009181005670 and displays the result:

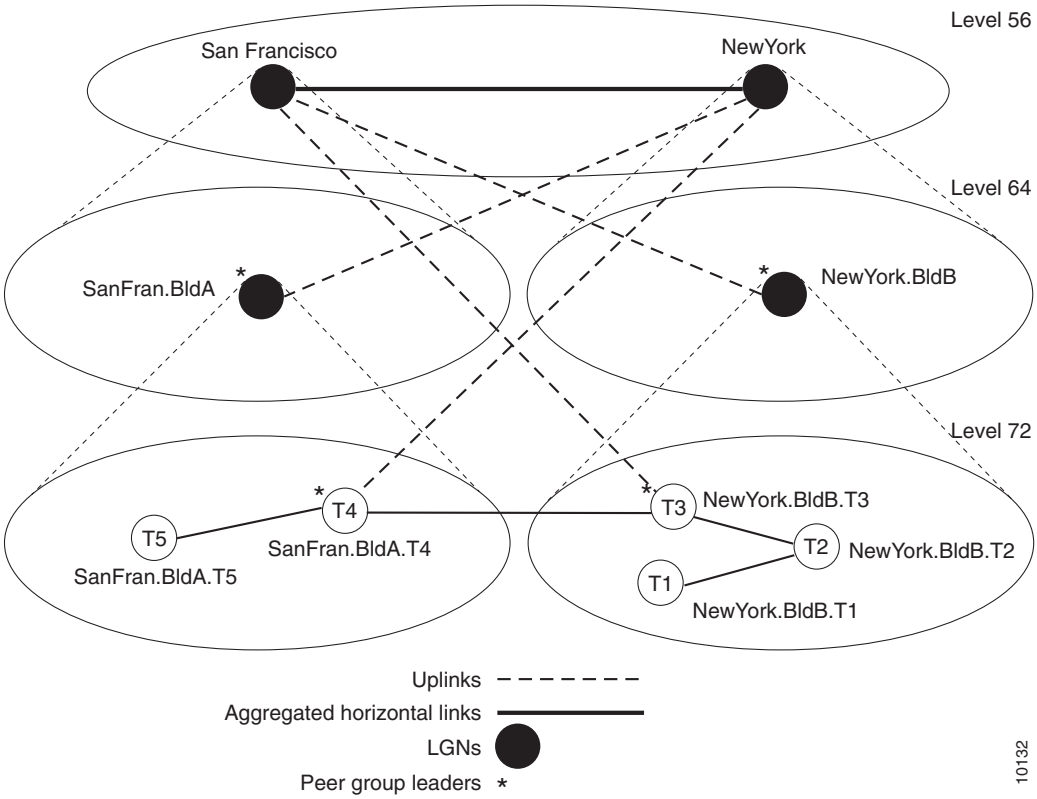
```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# node 1
```

```
DSLAM(config-pnni-node)# no auto-summary
DSLAM(config-pnni-node)# summary-address 47.009181005670
DSLAM# show atm pnni summary

Codes: Node - Node index advertising this summary
       Type - Summary type (INT - internal, EXT - exterior)
       Sup  - Suppressed flag (Y - Yes, N - No)
       Auto - Auto Summary flag (Y - Yes, N - No)
       Adv  - Advertised flag (Y - Yes, N - No)

 Node Type Sup Auto Adv  Summary Prefix
 ~~~~ ~~~~ ~~~ ~~~~ ~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  1   Int   N   Y    Y   47.0091.8100.0000.0040.0b0a.2a81/104
  2   Int   N   Y    N   47.01b1.0000.0000.0000.00/80
```

PNNI Hierarchy Configuration Example

An example configuration for a three-level hierarchical topology is shown in Figure 11-11. The example shows the configuration of only 5 switches, although you can configure several other switches in each peer group.

Figure 11-11 Example Three-Level Hierarchical Topology



At the lowest level (level 72), the hierarchy represents two separate peer groups. Each of the four switches named T2 to T5 are eligible to become a PGL at two levels, and each has two configured ancestor nodes (a parent node or a parent node's parent).

Switch T1 has no configured ancestor nodes and is not eligible to become a PGL. As a result of the peer group leader election at the lowest level, switches T4 and T3 become leaders of their peer groups. Therefore, each switch creates an LGN at the second level (level 64) of the hierarchy.

As a result of the election at the second level of the hierarchy, logical group nodes SanFran.BldA and NewYork.BldB are elected as PGLs, creating logical group nodes at the highest level of the hierarchy (Level 56). At that level, the uplinks induced through level 64 form an aggregated horizontal link within the common peer group at level 56.

Examples

The examples that follow show the configurations for each switch and the outputs of the **show atm pnni local-node** command.

Switch NewYork.BldB.T1 Configuration

```
hostname NewYork.BldB.T1
atm address 47.0091.4455.6677.1144.1011.1233.0060.3e7b.3a01.00
atm router pnni
  node 1 level 72 lowest
  redistribute atm-static

NewYork.BldB.T1# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: NewYork.BldB.T1
  System address          47.009144556677114410111233.00603E7B3A01.01
  Node ID          72:160:47.009144556677114410111233.00603E7B3A01.00
  Peer group ID        72:47.0091.4455.6677.1144.0000.0000
  Level 72, Priority 0 0, No. of interfaces 3, No. of neighbors 2
  Parent Node Index: NONE

<information deleted>
```

Switch NewYork.BldB.T2 Configuration

```
hostname NewYork.BldB.T2
atm address 47.0091.4455.6677.1144.1011.1244.0060.3e5b.bc01.00
atm router pnni
node 1 level 72 lowest
  parent 2
  redistribute atm-static
  election leadership-priority 40
 node 2 level 64
  parent 3
  election leadership-priority 40
  name NewYork.BldB
 node 3 level 56
  name NewYork

NewYork.BldB.T2# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: NewYork.BldB.T2
  System address          47.009144556677114410111244.00603E5BBC01.01
  Node ID          72:160:47.009144556677114410111244.00603E5BBC01.00
  Peer group ID        72:47.0091.4455.6677.1144.0000.0000
  Level 72, Priority 40 40, No. of interfaces 3, No. of neighbors 1
  Parent Node Index: 2

<information deleted>
```

```
PNNI node 2 is enabled and not running
  Node name: NewYork.BldB
  System address          47.009144556677114410111244.00603E5BBC01.02
  Node ID           64:72:47.009144556677114400000000.00603E5BBC01.00
  Peer group ID        64:47.0091.4455.6677.1100.0000.0000
  Level 64, Priority 40 40, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: 3

<information deleted>

PNNI node 3 is enabled and not running
  Node name: NewYork
  System address          47.009144556677114410111244.00603E5BBC01.03
  Node ID           56:64:47.009144556677110000000000.00603E5BBC01.00
  Peer group ID        56:47.0091.4455.6677.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: NONE

<information deleted>
```

Switch NewYork.BldB.T3 Configuration

```
hostname NewYork.BldB.T3
atm address 47.0091.4455.6677.1144.1011.1255.0060.3e5b.c401.00
atm router pnni
 node 1 level 72 lowest
  parent 2
  redistribute atm-static
  election leadership-priority 45
 node 2 level 64
  parent 3
  election leadership-priority 45
  name NewYork.BldB
 node 3 level 56
  name NewYork

NewYork.BldB.T3# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: NewYork.BldB.T3
  System address          47.009144556677114410111255.00603E5BC401.01
  Node ID          72:160:47.009144556677114410111255.00603E5BC401.00
  Peer group ID        72:47.0091.4455.6677.1144.0000.0000
  Level 72, Priority 45 95, No. of interfaces 4, No. of neighbors 1
  Parent Node Index: 2

<information deleted>

PNNI node 2 is enabled and running
  Node name: NewYork.BldB
  System address          47.009144556677114410111255.00603E5BC401.02
  Node ID           64:72:47.009144556677114400000000.00603E5BC401.00
  Peer group ID        64:47.0091.4455.6677.1100.0000.0000
  Level 64, Priority 45 95, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: 3

<information deleted>

PNNI node 3 is enabled and running
  Node name: NewYork
  System address          47.009144556677114410111255.00603E5BC401.03
  Node ID           56:64:47.009144556677110000000000.00603E5BC401.00
```

```
  Peer group ID        56:47.0091.4455.6677.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 0, No. of neighbors 1
  Parent Node Index: NONE

<information deleted>
```

Switch SanFran.BldA.T4 Configuration

```
hostname SanFran.BldA.T4
atm address 47.0091.4455.6677.2233.1011.1266.0060.3e7b.2001.00
atm router pnni
 node 1 level 72 lowest
  parent 2
  redistribute atm-static
  election leadership-priority 45
 node 2 level 64
  parent 3
  election leadership-priority 45
  name SanFran.BldA
 node 3 level 56
  name SanFran

SanFran.BldA.T4# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: SanFran.BldA.T4
  System address          47.009144556677223310111266.00603E7B2001.01
  Node ID          72:160:47.009144556677223310111266.00603E7B2001.00
  Peer group ID        72:47.0091.4455.6677.2233.0000.0000
  Level 72, Priority 45 95, No. of interfaces 4, No. of neighbors 1
  Parent Node Index: 2

<information deleted>

PNNI node 2 is enabled and running
  Node name: SanFran.BldA
  System address          47.009144556677223310111266.00603E7B2001.02
  Node ID           64:72:47.009144556677223300000000.00603E7B2001.00
  Peer group ID        64:47.0091.4455.6677.2200.0000.0000
  Level 64, Priority 45 95, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: 3
<information deleted>

PNNI node 3 is enabled and running
  Node name: SanFran
  System address          47.009144556677223310111266.00603E7B2001.03
  Node ID           56:64:47.009144556677220000000000.00603E7B2001.00
  Peer group ID        56:47.0091.4455.6677.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 0, No. of neighbors 1
  Parent Node Index: NONE

<information deleted>
```

Switch SanFran.BldA.T5 Configuration

```
hostname SanFran.BldA.T5
atm address 47.0091.4455.6677.2233.1011.1244.0060.3e7b.2401.00
atm router pnni
 node 1 level 72 lowest
  parent 2
  redistribute atm-static
  election leadership-priority 10
 node 2 level 64
```

```
  parent 3
  election leadership-priority 40
  name SanFran.BldA
 node 3 level 56
  name SanFran

SanFran.BldA.T5# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: SanFran.BldA.T5
  System address          47.009144556677223310111244.00603E7B2401.01
  Node ID          72:160:47.009144556677223310111244.00603E7B2401.00
  Peer group ID        72:47.0091.4455.6677.2233.0000.0000
  Level 72, Priority 10 10, No. of interfaces 2, No. of neighbors 1
  Parent Node Index: 2

<information deleted>

PNNI node 2 is enabled and not running
  Node name: SanFran.BldA
  System address          47.009144556677223310111244.00603E7B2401.02
  Node ID           64:72:47.009144556677223300000000.00603E7B2401.00
  Peer group ID        64:47.0091.4455.6677.2200.0000.0000
  Level 64, Priority 40 40, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: 3

<information deleted>

PNNI node 3 is enabled and not running
  Node name: SanFran
  System address          47.009144556677223310111244.00603E7B2401.03
  Node ID           56:64:47.009144556677220000000000.00603E7B2401.00
  Peer group ID        56:47.0091.4455.6677.0000.0000.0000
  Level 56, Priority 0 0, No. of interfaces 0, No. of neighbors 0
  Parent Node Index: NONE

<information deleted>
```

Advanced PNNI Configuration

This section describes how to configure advanced PNNI features. The advanced features described in this section are not required to enable PNNI, but are provided to assist you in tuning your network performance.

Tuning Route Selection

This section describes how to tune the route selection in your PNNI network:

- Configuring Background Route Computation
- Configuring Link Selection
- Configuring the Maximum Administrative Weight Percentage
- Configuring the Precedence

Configuring Background Route Computation

The DSLAM supports these route selection modes:

- On-demand—A separate route computation is performed each time a SETUP or ADD PARTY message is received over a UNI or IISP interface. In this mode, the most recent topology information received by this node is always used for each setup request.
- Background routes—You can route calls using precomputed routing trees. In this mode, multiple background trees are precomputed for several service categories and QoS metrics. If no route can be found in the multiple background trees that satisfies the QoS requirements of a particular call, route selection reverts to on-demand route computation.

The background routes mode should be enabled in large networks where it will usually exhibit less-stringent processing requirements and better scalability. Route computation is performed at almost every poll interval when a significant change in the topology of the network is reported or when significant threshold changes have occurred since the last route computation.

To configure the background route computation, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**background-routes-enable** {**insignificant-threshold** *value* | **poll-interval** *seconds*}	Enable background routes and configure background route parameters.

Examples

This example shows how to enable background routes and configures the background routes poll interval to 30 seconds:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# background-routes-enable poll-interval 30
```

This example shows the ATM PNNI background route configuration using the **show atm pnni background status** privileged EXEC command:

```
DSLAM# show atm pnni background status

Background Route Computation is Enabled
Background Interval is set at 10 seconds
Background Insignificant Threshold is set at 32
```

This example shows the ATM PNNI background route tables for CBR using the **show atm pnni background routes** privileged EXEC command:

```
DSLAM# show atm pnni background routes cbr

  Background Routes From CBR/AW Table
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  2 Routes To Node 2
      1. Hops 1. 1:ATM0/2 -> 2
          ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
          <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
```

```
        2. Hops 1. 1:ATM0/1 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    1 Routes To Node 5
        1. Hops 1. 1:ATM1/0 -> 5
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    Background Routes From CBR/CDV Table
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    2 Routes To Node 2
        1. Hops 1. 1:ATM0/2 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
        2. Hops 1. 1:ATM0/1 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    1 Routes To Node 5
        1. Hops 1. 1:ATM0/1 -> 5
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    Background Routes From CBR/CTD Table
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  2 Routes To Node 2
        1. Hops 1. 1:ATM0/1 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
        2. Hops 1. 1:ATM0/1 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    1 Routes To Node 5
        1. Hops 1. 1:ATM0/1 -> 5
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    Background Routes From CBR/CTD Table
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    2 Routes To Node 2
        1. Hops 1. 1:ATM0/1 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
        2. Hops 1. 1:ATM0/2 -> 2
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10

    1 Routes To Node 5
        1. Hops 1. 1:ATM0/1 -> 5
            ->: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
            <-: aw 5040  cdv 138  ctd 154 acr 147743 clr0 10 clr01 10
```

Configuring Link Selection

The link selection feature allows you to choose the mode for selecting one specific link among several parallel links to the same neighbor node (for example, links between two adjacent switches).

When multiple parallel links are configured inconsistently, the order of precedence of configured values is as follows:

1. Admin-weight-minimize
2. Blocking-minimize
3. Transmit-speed-maximize
4. Load-balance

For example, if any link is configured as admin-weight minimize, that link is used for the entire link group.

To configure the PNNI link selection for, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.
2	**atm pnni link-selection {admin-weight-minimize | blocking-minimize | load-balance | transmit-speed-maximize}**	Configure ATM PNNI link selection for a specific link.

Example

This example shows how to configure ATM interface 0/0 to use the transmit-speed-maximize link selection mode and displays the result:

```
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# atm pnni link-selection transmit-speed-maximize
DSLAM# show atm pnni neighbor

  Neighbor Name: eng_22, Node number: 2
  Neighbor Node Id: 56:160:47.0091810000000003DDE74601.0003DDE74601.00
  Neighboring Peer State: Full
  Link Selection Set To: minimize blocking of future calls
    Port          Remote port ID          Hello state
    ATM0/1        ATM1/2    (81902000)  2way_in
    ATM0/2        ATM1/0    (81901000)  2way_in (Flooding Port)
```

Configuring the Maximum Administrative Weight Percentage

The maximum AW percentage feature allows you to prevent the use of alternate routes that consume too many network resources. This feature provides a generalized form of a hop count limit. The maximum acceptable administrative weight is equal to the specified percentage of the least administrative weight of any route to the destination (from the background routing tables). For example, if the least administrative weight to the destination is 5040 and the configured percentage is 300, the maximum acceptable administrative weight for the call is 5040 * 300 / 100 or 15120.

To configure the maximum AW percentage, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**max-admin-weight-percentage** *percentage*	Configure the maximum AW percentage. The value can range from 100 to 2000.

Note The **max-admin-weight-percentage** command takes effect only if background route computation is enabled.

Example

This example shows how to configure the node maximum AW percentage value as 300 and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# max-admin-weight-percentage 300
DSLAM# show atm pnni local-node
PNNI node 1 is enabled and running
  Node name: eng_1
  System address 47.009181000000000000001212.121212121212.00
  Node ID 56:160:47.009181000000000000001212.121212121212.00
  Peer group ID 56:47.0091.8100.0000.0000.0000.0000
  Level 56, Priority 0, No. of interface 4, No. of neighbor 1

  Hello interval 15 sec, inactivity factor 5, Hello hold-down 10 tenths of sec
  Ack-delay 2 sec, retransmit interval 10 sec, rm-poll interval 10 sec
  PTSE refresh interval 90 sec, lifetime factor 7, minPTSEinterval 1000 msec
  Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
  Default administrative weight mode: linespeed
   Max admin weight percentage: 300
  Next RM poll in 3 seconds
```

Configuring the Precedence

The route selection algorithm chooses routes to particular destinations using the longest match reachable address prefixes known to the DSLAM. When there are multiple longest match reachable address prefixes known to the DSLAM, the route selection algorithm first attempts to find routes to reachable addresses with types of greatest precedence. Among multiple longest match reachable address prefixes of the same type, routes with the least total AW are chosen first.

Local internal reachable addresses, whether learned through ILMI or as static routes, receive highest precedence or a precedence value of one. The precedence of other reachable address types is configurable.

To configure the precedence of reachable addresses, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**precedence** [**pnni-remote-exterior** *value_2-4* | **pnni-remote-exterior-metrics** *value_2-4* | **pnni-remote-internal** *value_2-4* | **pnni-remote-internal-metrics** *value_2-4* | **static-local-exterior** *value_2-4* | **static-local-exterior-metrics** *value_2-4* | **static-local-internal-metrics** *value_2-4*]	At the configure ATM router prompt, enter PNNI precedence and configure the PNNI node.

Example

This example shows how to configure all PNNI remote exterior routes with a precedence value of 4 and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# precedence pnni-remote-exterior 4
DSLAM# show atm pnni precedence
                                Working   Default
  Prefix Poa Type               Priority  Priority
  ----------------------------  --------  --------
  local-internal                   1         1
  static-local-internal-metrics    2         2
  static-local-exterior            3         3
  static-local-exterior-metrics    2         2
  pnni-remote-internal             2         2
  pnni-remote-internal-metrics     2         2
  pnni-remote-exterior             4         4
  pnni-remote-exterior-metrics     2         2
```

Tuning Topology Attributes

This section describes how to configure attributes that affect the network topology.

Configuring the Global Administrative Weight Mode

Administrative weight is the primary routing metric for minimizing use of network resources. You can configure the administrative weight (AW) to indicate the relative desirability of using a link. In addition to the per-interface **atm pnni administrative-weight** command, the ATM router PNNI **administrative-weight** command can be used to change the default AW assignment. For example, you can assign equal AWs to all links in the network to minimize the number of hops used by each connection.

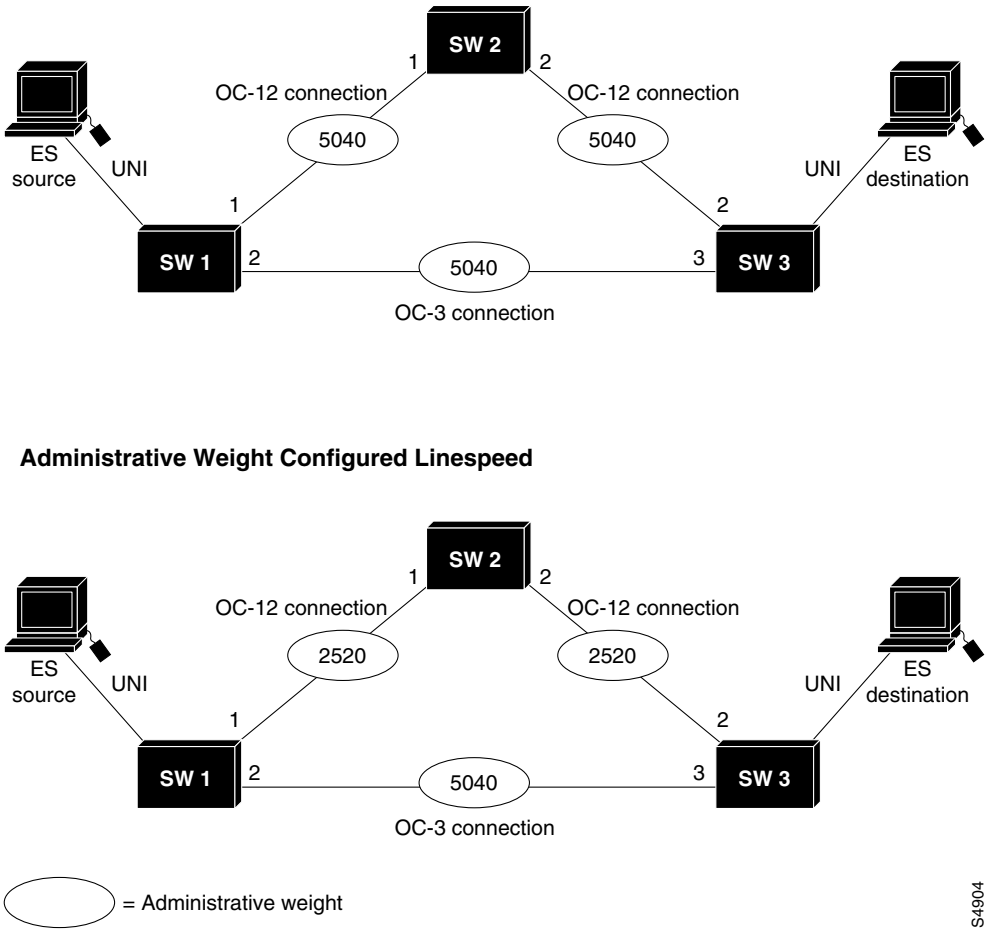
To configure the administrative weight mode, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**administrative-weight** {**linespeed** | **uniform**}	At the configure router prompt, configure the administrative weight for all node connections.

Figure 11-12 is an example of the effect of AW on call routing. The network depicted at the top of Figure 11-12 is configured as uniform, causing equal AW to be assigned to each link. The identical network at the bottom of the figure is configured as line speed.

The links between SW1 and SW2 (SW1p1 to SW2p1) and SW2 and SW3 (SW2p2 to SW3p2) are both faster OC-12 connections and have lower AWs. PNNI interprets the route over the two OC-12 links as being administratively equivalent to a more direct route between SW1 and SW3 using the OC-3 connection.

Figure 11-12 Network Administrative Weight Example



Example

This example shows how to configure AW for the node as line speed and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# administrative-weight linespeed
DSLAM# show atm pnni local-node

PNNI node 1 is enabled and running
  Node name: DSLAM
  System address 47.009181000000000000001212.121212121212.00
  Node ID 56:160:47.009181000000000000001212.121212121212.00
  Peer group ID 56:47.0091.8100.0000.0000.0000.0000
  Level 56, Priority 0, No. of interface 4, No. of neighbor 1
   Hello interval 15 sec, inactivity factor 5, Hello hold-down 10 tenths of sec
  Ack-delay 2 sec, retransmit interval 10 sec, rm-poll interval 10 sec
  PTSE refresh interval 90 sec, lifetime factor 7, minPTSEinterval 1000 msec
  Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
   Default administrative weight mode: linespeed
  Max admin weight percentage: 300
  Next RM poll in 3 seconds
```

Configuring Administrative Weight per Interface

AW is the main metric used for computation of the paths by PNNI. The assignment of AWs to links and nodes affects the way PNNI selects paths in the private ATM network.

To configure the administrative weight on an interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.
2	**atm pnni admin-weight** *number traffic_class*	Configure the ATM AW for this link.

Example

This example shows how to configure ATM interface 0/0 with ATM PNNI AW of 7560 for traffic class ABR and displays the result:

```
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# atm pnni admin-weight 7560 abr
DSLAM# show atm pnni interface atm 0/0 detail

Port ATM0/0 is up  , Hello state 2way_in with node eng_18
    Next hello occurs in 11 seconds, Dead timer fires in 73 seconds
    CBR    : AW 5040 MCR 155519 ACR 147743 CTD 154 CDV 138 CLR0 10 CLR01 10
    VBR-RT : AW 5040 MCR 155519 ACR 155519 CTD 707 CDV 691 CLR0 8 CLR01 8
    VBR-NRT: AW 5040 MCR 155519 ACR 155519 CLR0 8 CLR01 8
     ABR    : AW 7560 MCR 155519 ACR 0
    UBR    : AW 5040 MCR 155519
    Remote node ID 56:160:47.00918100000000613E7B2F01.00613E7B2F99.00
    Remote node address 47.00918100000000613E7B2F01.00613E7B2F99.00
    Remote port ID  ATM0/1   (80102000) (0)
```

Configuring Transit Restriction

Transit calls originate from another ATM DSLAM and pass through the DSLAM. You may want to set your edge switches to eliminate this transit traffic and only allow traffic originating or terminating at the switch.

To configure a transit restriction, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**transit-restricted**	Enable transit restricted on this node.

Example

This example shows how to enable the transit-restricted feature and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# node 1
DSLAM(config-pnni-node)# transit-restricted
DSLAM# show atm pnni local-node

   PNNI node 1 is enabled and running
    Node name: DSLAM
    System address 47.00918100000000400B0A3081.00400B0A3081.00
    Node ID 56:160:47.00918100000000400B0A3081.00400B0A3081.00
    Peer group ID 56:47.0091.8100.0000.0000.0000.0000
    Level 56, Priority 0, No. of interfaces 4, No. of neighbors 2
     Node Does Not Allow Transit Calls

    Hello interval 15 sec, inactivity factor 5,
    Hello hold-down 10 tenths of sec
    Ack-delay 10 tenths of sec, retransmit interval 5 sec,
    Resource poll interval 5 sec
    PTSE refresh interval 1800 sec, lifetime factor 200 percent,
    Min PTSE interval 10 tenths of sec
    Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
    Default administrative weight mode: uniform
    Max admin weight percentage: -1
    Next resource poll in 3 seconds
    Max PTSEs requested per PTSE request packet: 32
    Redistributing static routes: Yes
```

Configuring Redistribution

Redistribution instructs PNNI to distribute reachability information from non-PNNI sources throughout the PNNI routing domain. The DSLAM supports redistribution of static routes, such as those configured on IISP interfaces.

Note By default, redistribution of static routes is enabled.

To enable redistribution of static routes, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**redistribute atm-static**	Enable redistribution of static routes.

Example

This example shows how to enable redistribution of static routes and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# node 1
DSLAM(config-pnni-node)# redistribute atm-static
DSLAM# show atm pnni local-node
  PNNI node 1 is enabled and running
    Node name: DSLAM
    System address 47.00918100000000400B0A3081.00400B0A3081.00
    Node ID 56:160:47.00918100000000400B0A3081.00400B0A3081.00
    Peer group ID 56:47.0091.8100.0000.0000.0000.0000
    Level 56, Priority 0, No. of interfaces 4, No. of neighbors 2
    Node Allows Transit Calls

    Hello interval 15 sec, inactivity factor 5,
    Hello hold-down 10 tenths of sec
    Ack-delay 10 tenths of sec, retransmit interval 5 sec,
    Resource poll interval 5 sec
    PTSE refresh interval 1800 sec, lifetime factor 200 percent,
    Min PTSE interval 10 tenths of sec
    Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
    Default administrative weight mode: uniform
    Max admin weight percentage: -1
    Next resource poll in 3 seconds
    Max PTSEs requested per PTSE request packet: 32
     Redistributing static routes: Yes
```

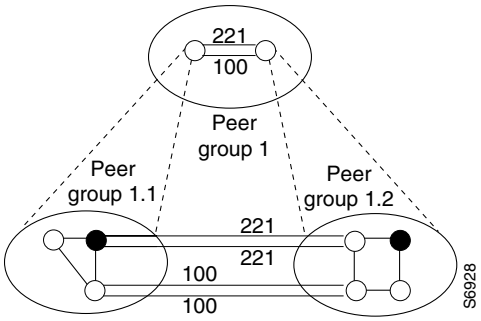
Configuring Aggregation Token

One of the tasks performed by the LGN is link aggregation. These terms describe the link aggregation algorithms:

- An uplink is a link to a higher level node, called an upnode.
- The term higher means at a higher level in the hierarchy compared to the level of our peer group.
- The aggregation token controls the grouping of multiple physical links into logical links.
- Uplinks to the same upnode, with the same aggregation token value, are represented at a higher level as horizontal aggregated links.
- Resource Availability Information Groups (RAIGs) are computed according to the aggregation algorithm.

Figure 11-13 shows four physical links between four ATM switches. Two physical links between two ATM switches in different PGs are assigned the PNNI aggregation token value of 221; the other two are assigned the value of 100. These lines are summarized and represented in the next higher PNNI level.

Figure 11-13 PNNI Aggregation Token



When you configure the PNNI aggregation token

- You only need to configure the interface on only one side of the link. If the configured aggregation token value of one side is zero and the other side is non-zero, the non-zero value is used by both sides as the aggregation token value.
- If you choose to configure an aggregation token value on both interfaces, make sure the aggregation token values match. If the values do not match, the configuration is invalid and the default aggregation token value of zero is used.

To specify an aggregation token value, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**interface atm** *slot/port*	Specify the ATM interface.
2	**atm pnni aggregation-token** *value*	Enter a value for the aggregation-token on the ATM interface.

Example

This example shows how to configure an aggregation token on ATM interface 0/2 and displays the result (note that the show command includes the detail keyword):

```
DSLAM(config)# interface atm 0/2
DSLAM(config-if)# atm pnni aggregation-token 100
NewYork.BldB.T3 # show atm pnni interface atm0/2 detail

PNNI Interface(s) for local-node 1 (level=56):

Port ATM0/2 RCC is up  , Hello state common_out with node SanFran.BldA.T4
  Next hello occurs in 4 seconds, Dead timer fires in 72 seconds
  CBR    : AW 5040 MCR 155519 ACR 147743 CTD 154 CDV 138 CLR0 10 CLR01 10
  VBR-RT : AW 5040 MCR 155519 ACR 155519 CTD 707 CDV 691 CLR0 8 CLR01 8
  VBR-NRT: AW 5040 MCR 155519 ACR 155519 CLR0 8 CLR01 8
  ABR    : AW 5040 MCR 155519 ACR 0
  UBR    : AW 5040 MCR 155519
  Aggregation Token: configured 0 , derived 2, remote 2
  Tx ULIA seq# 1, Rx ULIA seq# 1, Tx NHL seq# 1, Rx NHL seq# 2
  Remote node ID        72:160:47.009144556677223310111266.00603E7B2001.00
  Remote node address          47.009144556677223310111266.00603E7B2001.01
  Remote port ID         ATM0/0  (80003000) (0)
```

```
  Common peer group ID      56:47.0091.4455.6677.0000.0000.0000
  Upnode ID              56:72:47.009144556677223300000000.00603E7B2001.00
  Upnode Address               47.009144556677223310111266.00603E7B2001.02
  Upnode number: 11       Upnode Name: SanFran
NewYork.BldB.T3#
```

Configuring the Aggregation Mode

The DSLAM has two algorithms to perform link aggregation:

- Best link—Selects a single optimal uplink, based on a selected parameter, and assigns the aggregated RAIG based on that uplink. With this aggregation algorithm, there is always a link that has the advertised RAIG parameters. The default aggregation mode is best link.
- Aggressive—Examines each RAIG parameter and selects the best (optimal) value over all aggregated links. This procedure is repeated for each parameter. The resulting aggregated parameters reflect a best case that might not be represented by an existing uplink. Such an algorithm tends to attract calls towards the aggregated link. Because it might overestimate the available resources, it is termed aggressive.

All interfaces default to an aggregation token value of zero, so that by default all parallel outside links between a pair of peer groups are aggregated at higher levels. If the metrics of the various parallel outside links differ by very large ratios, you can improve the routing accuracy by assigning a different aggregation token to some links so that PNNI routing considers them separately at the higher levels.

To configure the aggregation mode for a traffic class, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	Enter ATM router PNNI mode from the terminal.
2	**node** *node_index*	Enter node configuration mode.
3	**aggregation-mode** {**link**} *traffic-class* {**best-link** | **aggressive**}	Configure the aggregation mode for a specific service category (traffic class).

Example

This example shows how to configure aggressive link aggregation mode for CBR traffic and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-pnni-node)# node 2
DSLAM(config-pnni-node)# aggregation-mode link cbr aggressive
DSLAM# show atm pnni aggregation link

PNNI PGL link aggregation for local-node 2 (level=72, name=DSLAM.2.72)

  Configured aggregation modes (per service class):
     CBR         VBR-RT       VBR-NRT       ABR           UBR
  ~~~~~~~~~~~  ~~~~~~~~~~  ~~~~~~~~~~  ~~~~~~~~~~  ~~~~~~~~~~
  aggressive   best-link    best-link    best-link    best-link

 No Aggregated links for this node.
```

Configuring Significant Change Thresholds

PNNI topology state packets (PTSEs) can overwhelm the network if they are transmitted each time a parameter in the network changes. To avoid this problem, PNNI uses significant change thresholds that control the origination of PTSEs.

Note Any change in AW and CLR is considered significant and triggers a new PTSE.

To configure the PTSE significant change threshold, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**ptse significant-change** {**acr-mt** *percentage* | **acr-pm** *multiplier* | **cdv-pm** *multiplier* | **ctd-pm** *multiplier*}	Configure a PTSE significant change percentage or multiplier.

Example

This example shows how to configure a PTSE with a significant change percentage of 30, and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# node 1
DSLAM(config-pnni-node)# ptse significant-change acr-pm 30
DSLAM# show atm pnni resource-info

PNNI:80.1 Insignificant change parameters
acr pm 50,  acr mt 3, cdv pm 25, ctd pm 50, resource poll interval 5 sec
Interface insignificant change bounds:
Interface ATM0/1
  CBR    : MCR 155519, ACR 147743 [73871,366792], CTD 50 [25,75],CDV 34 [26,42],
 CLR0 10, CLR01 10,
  VBR-RT : MCR 155519, ACR 155519 [77759,366792], CTD 359 [180,538],CDV 342 [257
,427], CLR0 8, CLR01 8,
  VBR-NRT: MCR 155519, ACR 155519 [77759,155519], CLR0 8, CLR01, 8
  ABR    : MCR 155519 ACR 147743 [73871,155519]
  UBR    : MCR 155519
Interface ATM1/0
  CBR    : MCR 155519, ACR 147743 [73871,366792], CTD 50 [25,75],CDV 34 [26,42],
 CLR0 10, CLR01 10,
  VBR-RT : MCR 155519, ACR 155519 [77759,366792], CTD 359 [180,538],CDV 342 [257
,427], CLR0 8, CLR01 8,
  VBR-NRT: MCR 155519, ACR 155519 [77759,155519], CLR0 8, CLR01, 8
  ABR    : MCR 155519 ACR 147743 [73871,155519]
  UBR    : MCR 155519
<information deleted>
```

Tuning Protocol Parameters

This section describes how to tune the PNNI protocol parameters.

Configuring PNNI Hello, Database Synchronization, and Flooding Parameters

PNNI uses the Hello protocol to determine the status of neighbor nodes, and uses PTSEs to disseminate topology database information in the ATM network.

To configure the Hello protocol parameters and PTSE significant change, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**node** *node_index*	At the configure ATM router prompt, enter node configuration mode. The prompt changes to DSLAM(config-pnni-node)#.
3	**timer** [**ack-delay** *tenths_of_seconds*] [**hello-holddown** *tenths_of_seconds*] [**hello-interval** *seconds*] [**inactivity-factor** *number*] [**retransmit-interval** *seconds*]	Configure Hello database synchronization and flooding parameters.
4	**ptse** [**lifetime-factor** *percentage_factor*] [**min-ptse-interval** *tenths_of_seconds*] [**refresh-interval** *seconds*] [**request** *number*] [**significant-change** *acr-mt percent*] [**significant-change** *acr-pm percent*] [**significant-change** *cdv-pm percent*] [**significant-change** *ctd-pm percent*]	Configure PTSE significant change percent number.

Example

This example shows how to configure the PTSE refresh interval to 600 seconds:

```
DSLAM(config-pnni-node)# ptse refresh-interval 600
```

This example shows how to configure the retransmission of the Hello timer to 60 seconds:

```
DSLAM(config-pnni-node)# timer hello-interval 60
```

This example shows the ATM PNNI Hello, database synchronization, and flooding configuration using the **show atm pnni local-node** privileged EXEC command:

```
DSLAM# show atm pnni local-node
PNNI node 1 is enabled and running
    Node name: DSLAM
    System address 47.00918100000000400B0A3081.00400B0A3081.00
    Node ID 56:160:47.00918100000000400B0A3081.00400B0A3081.00
    Peer group ID 56:47.0091.8100.0000.0000.0000.0000
    Level 56, Priority 0, No. of interfaces 4, No. of neighbors 2
    Node Allows Transit Calls
```

```
  Hello interval 60 sec, inactivity factor 5,
Hello hold-down 10 tenths of sec
Ack-delay 10 tenths of sec, retransmit interval 5 sec,
Resource poll interval 5 sec
  PTSE refresh interval 600 sec, lifetime factor 200 percent,
Min PTSE interval 10 tenths of sec
Auto summarization: on, Supported PNNI versions: newest 1, oldest 1
Default administrative weight mode: uniform
Max admin weight percentage: -1
Next resource poll in 3 seconds
Max PTSEs requested per PTSE request packet: 32
Redistributing static routes: Yes
```

Configuring the Resource Management Poll Interval

The resource management poll interval specifies the frequency with which PNNI polls resource management to update the values of link metrics and attributes. You can configure the resource poll interval to control the trade-off between the processing load and the accuracy of PNNI information. A larger value will probably generate a smaller number of PTSE updates. A smaller value results in greater accuracy in tracking resource information.

To configure the resource management poll interval, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**resource-poll-interval** *seconds*	Configure the resource management poll interval.

Example

This example configures the RM poll interval to 10 seconds and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# resource-poll-interval 10
DSLAM# show atm pnni resource-info
PNNI:80.1 Insignificant change parameters
acr pm 50,  acr mt 3, cdv pm 25, ctd pm 50, resource poll interval 10 sec
Interface insignificant change bounds:
Interface ATM0/1
  CBR    : MCR 155519, ACR 147743 [73871,366792], CTD 50 [25,75],CDV 34 [26,42],
 CLR0 10, CLR01 10,
  VBR-RT : MCR 155519, ACR 155519 [77759,366792], CTD 359 [180,538],CDV 342 [257
<information deleted>
```

Configuring Statistics Collection

This section describes how to collect statistics about the routing of ATM connections.

To enable statistics collection, perform these steps, beginning in global configuration mode:

Step	Command	Task
1	**atm router pnni**	At the configure prompt, enter ATM router PNNI mode from the terminal. The prompt changes to DSLAM(config-atm-router)#.
2	**statistics** [**call**]	Enable ATM PNNI statistics gathering.

Example

This example shows how to enable PNNI ATM statistics gathering and displays the result:

```
DSLAM(config)# atm router pnni
DSLAM(config-atm-router)# statistics call
DSLAM# show atm pnni statistics call

pnni call statistics since 22:19:29

                    total     cbr       rtvbr     nrtvbr    abr       ubr
source route reqs   1346      0         0         0         0         0
successful          1342      1342      0         0         0         0
unsuccessful        4         4         0         0         0         0
crankback reqs      0         0         0         0         0         0
successful          0         0         0         0         0         0
unsuccessful        0         0         0         0         0         0
on-demand attempts  0         0         0         0         0         0
successful          0         0         0         0         0         0
unsuccessful        0         0         0         0         0         0
background lookups  0         0         0         0         0         0
successful          0         0         0         0         0         0
unsuccessful        0         0         0         0         0         0
next port requests  0         0         0         0         0         0
successful          0         0         0         0         0         0
unsuccessful        0         0         0         0         0         0

                    total     average
usecs in queue      2513166   1867
usecs in dijkstra   0         0
usecs in routing    132703    98
```

CHAPTER 12

Using Access Control

This chapter describes how to configure and maintain access control lists, which are used to permit or deny incoming calls or outgoing calls on an interfaces of Cisco DSLAMs with NI-2. This chapter includes these sections:

Access Control Overview

The ATM signaling software uses the access control list to filter setup messages on an interface based on destination, source, or a combination of both. You can use access lists to deny connections known to be security risks and permit all other connections, or to permit only those connections considered acceptable and deny all the rest. For firewall implementation, denying access to security risks offers more control.

During initial configuration, perform these steps to use access control to filter setup messages:

Step 1 Create a template alias allowing you to use real names instead of ATM addresses in your ATM filter expressions.

Step 2 Create the ATM filter set or filter expression based on your requirements.

Step 3 Associate the filter set or filter expression to an interface using the **atm access-group** command.

Step 4 Confirm the configuration.

Configuring a Template Alias

To configure an ATM template alias, use this command in global configuration mode:

Command	Task
atm template-alias *name template*	Configure a global ATM address template alias.

Examples

This example creates a template alias named *training* using the ATM address template 47.1328 and the ellipses (...) to enter the trailing 4-bit hexadecimal digits in the address:

```
DSLAM(config)# atm template-alias training 47.1328...
```

This example creates a template alias named *bit_set* with the ATM address template 47.9f9.(1*0*).88ab... that matches the 4 addresses that begin with

- 47.9F9(1*000*).88AB... = 47.9F9*8*.88AB...
- 47.9F9(1*001*).88AB... = 47.9F9*9*.88AB...
- 47.9F9(1*100*).88AB... = 47.9F9*C*.88AB...
- 47.9F9(1*101*).88AB... = 47.9F9*D*.88AB...

```
DSLAM(config)# atm template-alias bit_set 47.9f9(1*0*).88ab...
```

This example creates a template alias named *byte_wise* with the ATM address template 47.9*F8.33... that matches all ATM addresses beginning with the 16 prefixes:

- 47.9*0*F8.33...

  through

- 47.9*F*F8.33...

```
DSLAM(config)# atm template-alias byte_wise 47.9*F8.33...
```

This example shows the template aliases configured in the previous examples using the **show running-config** privileged EXEC command:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!
version XX.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
!
username dtate
ip rcmd remote-username dplatz
atm template-alias training 47.1328...
atm template-alias bit_set 47.9f9(1*0*).88ab...
atm template-alias byte_wise 47.9*f8.33...
!
<information deleted>
```

Configuring ATM Filter Sets

To create an ATM address filter or time-of-day filter, use this command in global configuration mode:

Command	Task
atm filter-set *name* [**index** *number*] [**permit** | **deny**] {*address-template* | **time-of-day** {**anytime** | *start-time end-time*}}	Configure a global ATM address filter set.

Examples

This example creates a filter named *filter_1* that permits access to the specific ATM address 47.0000.8100.1234.0003.c386.b301.0003.c386.b301.00:

```
DSLAM(config)# atm filter-set filter_1 permit 47.0000.8100.1234.0003.c386.b301.0003.c386.b301.00
```

This example creates a filter named *filter_2* that denies access to the specific ATM address 47.000.8100.5678.0003.c386.b301.0003.c386.b301.00, but allows access to all other ATM addresses:

```
DSLAM(config)# atm filter-set filter_2 deny 47.0000.8100.5678.0003.c386.b301.0003.c386.b301.00
DSLAM(config)# atm filter-set filter_2 permit default
```

This example creates a filter named *filter_3* that denies access to all ATM addresses that begin with the prefix 47.840F, but permits all other calls:

```
DSLAM(config)# atm filter-set filter_3 deny 47.840F...
DSLAM(config)# atm filter-set filter_3 permit default
```

Note The order in which deny and permit filters are configured is very important. See the next example.

In this example, the first filter set, *filter_4*, has its first filter configured to permit all addresses and its second filter configured to deny access to all addressees that begin with the prefix 47.840F. Since the default filter matches all addresses, the second filter is never used. Addresses that begin with prefix 47.840F are also permitted.

```
DSLAM(config)# atm filter-set filter_4 permit default
DSLAM(config)# atm filter-set filter_4 deny 47.840F...
```

This example creates a filter named *filter_5* that denies access to all ATM addresses described by the ATM template alias bad_users:

```
DSLAM(config)# atm filter-set filter_5 deny bad_users
DSLAM(config)# atm filter-set filter_5 permit default
```

This example shows how to configure a filter set named *tod1*, with an index of 2, to deny calls between 11:15 a.m. and 10:45 p.m.:

```
DSLAM(config)# atm filter-set tod1 index 2 deny time-of-day 11:15 22:45
DSLAM(config)# atm filter-set tod1 index 3 permit time-of-day anytime
```

This example shows how to configure a filter set named *tod1*, with an index of 4, to permit calls any time:

```
DSLAM(config)# atm filter-set tod1 index 4 permit time-of-day anytime
```

This example shows how to configure a filter set named *tod2* to deny calls between 6:00 a.m. and 8:00 p.m.:

```
DSLAM(config)# atm filter-set tod2 deny time-of-day 20:00 06:00
```

```
DSLAM(config)# atm filter-set tod2 permit time-of-day anytime
```

This example shows how to configure a filter set named *tod2* to permit calls at any time:

```
DSLAM(config)# atm filter-set tod2 permit time-of-day 3:30 3:30
```

After you create a filter set using the previous configuration commands, it must be associated with an interface as an access group to actually filter any calls (see the "Configuring ATM Interface Access Control" section on page 12-243).

Deleting Filter Sets

To delete an ATM filter set, use this command in global configuration mode

:

Command	Task
no atm filter-set *name* [**index** *number*]	Delete a global ATM address filter set.

Example

This example shows how to display and delete filter sets:

```
DSLAM# show atm filter-set
ATM filter set tod1
  deny From 11:15 Hrs Till 22:45 Hrs  index 2
  permit From 0:0 Hrs Till 0:0 Hrs  index 4
ATM filter set tod2
  deny From 20:0 Hrs Till 6:0 Hrs  index 1
  permit From 3:30 Hrs Till 3:30 Hrs  index 2
DSLAM# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)# no atm filter-set tod1 index 2
DSLAM(config)# no atm filter-set tod2
DSLAM(config)# end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM# show atm filter-set
ATM filter set tod1
  permit From 0:0 Hrs Till 0:0 Hrs  index 4
```

In order, the commands in this example:

1. Display the existing filter sets using the **show atm filter-set** command.
2. Change to EXEC configuration mode.
3. Delete the specific filter-set tod1 index 1.
4. Delete the entire filter-set tod2.
5. Display the modified filter sets using the **show atm filter-set** command.

Configuring an ATM Filter Expression

Use the following commands to create global ATM filter expressions in global configuration mode.

Command	Task
atm filter-expr *name term*	Define a simple filter expression with only one *term* and no operators.
atm filter-expr *name* [**destination** | **source** | **src**] *term1* **and** [**destination** | **source** | **src**] *term2*	Define a filter expression using the operator and.
atm filter-expr *name* [**destination** | **source** | **src**] **not** *term*	Define a filter expression using the operator not.
atm filter-expr *name* [**destination** | **source** | **src**] *term1* **or** [**destination** | **source** | **src**] *term2*	Define a filter expression using the operator or.
atm filter-expr *name* [**destination** | **source** | **src**] *term1* **xor** [**destination** | **source** | **src**] *term2*	Define a filter expression using the operator xor.
no atm filter-expr *name*	Delete a filter.

Examples

This example defines a simple filter expression that has only one term and no operators:

```
DSLAM(config)# atm filter-expr training filter_1
```

This example defines a filter expression using the **not** operator:

```
DSLAM(config)# atm filter-expr training not filter_1
```

This example defines a filter expression using the **or** operator:

```
DSLAM(config)# atm filter-expr training filter_2 or filter_1
```

This example defines a filter expression using the **and** operator:

```
DSLAM(config)# atm filter-expr training filter_1 and source filter_2
```

This example defines a filter expression using the **xor** operator:

```
DSLAM(config)# atm filter-expr training filter_2 xor filter_1
```

Configuring ATM Interface Access Control

To subscribe an ATM interface to an existing ATM filter set or filter expression, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm access-group** *name* [**in** | **out**]	Configure an existing ATM address pattern matching the filter expression.

Examples

This example shows how to configure access control for outgoing calls on ATM interface 0/1:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm access-group training out
```

This example configures access control for both outgoing and incoming calls on ATM interface 0/1 and displays the configured ATM filters:

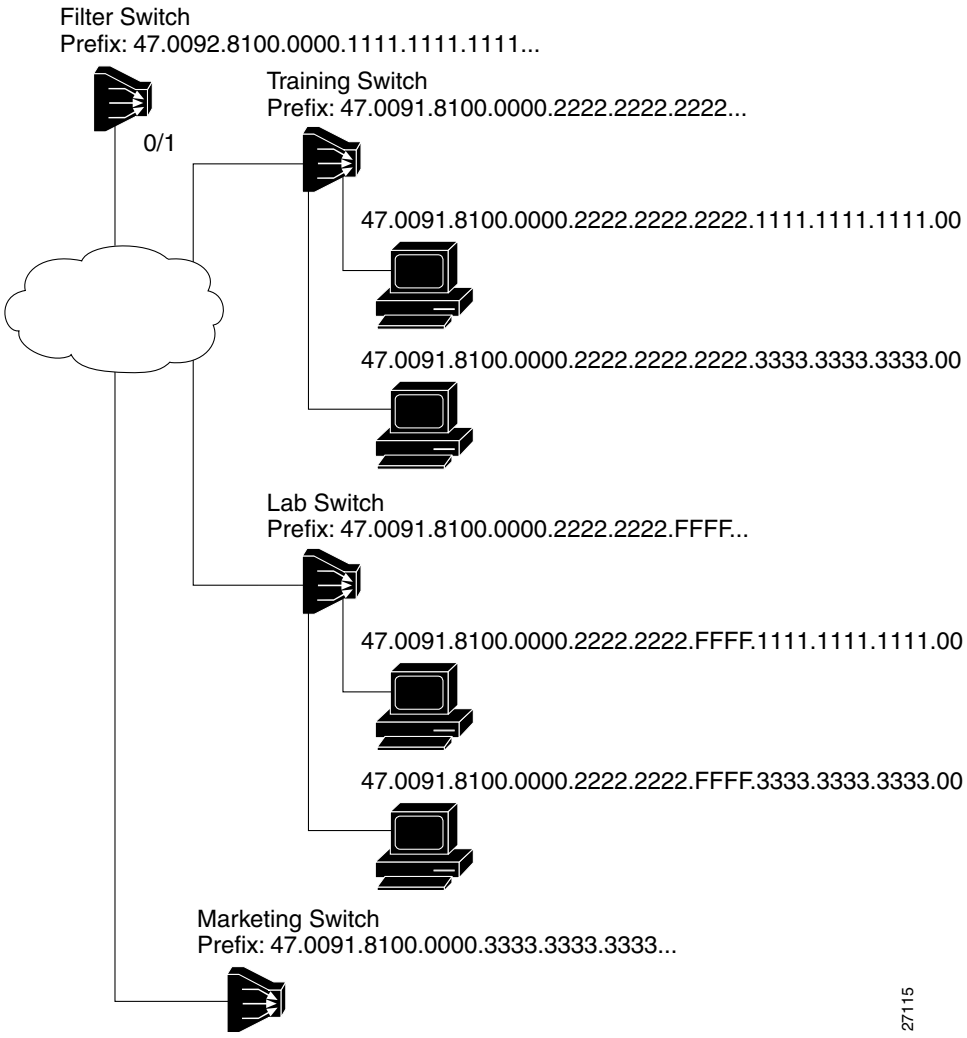
```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm access-group training out
DSLAM(config-if)# atm access-group marketing in

DSLAM# show atm filter-set
ATM filter set tod1
  deny From 11:15 Hrs Till 22:45 Hrs  index 2
  permit From 0:0 Hrs Till 0:0 Hrs  index 4
ATM filter set tod2
  deny From 20:0 Hrs Till 6:0 Hrs  index 1
  permit From 3:30 Hrs Till 3:30 Hrs  index 2

DSLAM# show atm filter-expr
training = dest filter_1
```

ATM Filter Configuration Example

This section provides a complete access filter configuration example using the information described in the preceding sections.

The sample network configuration used in this filter set configuration scenario is shown in Figure 12-1.

Figure 12-1 ATM Access Filter Configuration Example



Example

This example shows how to configure the Filter Switch, shown in Figure 12-1, to deny access to all calls received on ATM interface 0/1 from the workstations directly attached to the Lab Switch, but to allow all other calls. The Filter Switch denies all calls if the calling party address begins with the prefix 47.0091.8100.0000.2222.2222.FFFF:

```
Filter Switch(config)# atm template-alias lab-sw 47.0091.8100.0000.2222.2222.FFFF...
Filter Switch(config)# atm filter-set filter_1 deny lab-sw
Filter Switch(config)# atm filter-set filter_1 permit default
Filter Switch(config)# atm filter-expr exp1 src filter_1
Filter Switch(config)#
Filter Switch(config)# interface atm 0/1
Filter Switch(config-if)# atm access-group exp1 in
Filter Switch(config-if)# end
Filter Switch# show atm filter-set
ATM filter set filter_1
  deny   47.0091.8100.0000.2222.2222.ffff...  index 1
  permit default  index 2
Filter Switch# show atm filter-expr
```

```
exp1 = src filter_1
```

Configuring Per-Interface Address Registration with Optional Access Filters

The DSLAM allows you to configure per-interface access filters for ILMI address registration to override the global default of access filters.

To configure ILMI address registration and the optional access filters for a specified interface, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.
2.	**atm address-registration permit** {**all** | **matching-prefix** [**all-groups** | **wellknown-groups**]}	Configure ILMI address registration and the optional access filters for a specified interface.

Example

This example shows how to configure ILMI address registration on an individual interface to permit all groups with a matching ATM address prefix and displays the interface ILMI address registration access filter configuration:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm address-registration permit matching-prefix all-groups
%ATM-5-ILMIACCFILTER: New access filter setting will be applied to registration
of new addresses on ATM0/1.
DSLAM(config-if)#

DSLAM# show running-config
Building configuration...
Current configuration:
!
version XX.X
no service pad

<Information Deleted>

interface ATM0/0
 no ip address
 atm maxvp-number 0
!
interface Ethernet0/0
 ip address 172.20.41.110 255.255.255.0
 ip access-group 102 out
!
interface ATM0/1
 no atm auto-configuration
 atm address-registration permit matching-prefix all-groups
 atm iisp side user
 atm pvc 100 200
 atm signalling cug access permit-unknown-cugs both-direction permanent
 atm accounting
!
```

```
interface ATM0/2
!

<information deleted>
```

CHAPTER 13

Configuring In-Band Management

This chapter describes how to configure in-band management on Cisco DSLAMs with NI-2.

This chapter includes the sections:

Configuring In-Band Management

The DSLAM allows in-band management via the trunk interface. In-band management uses the IP over ATM protocol. The DSLAM is a client in an IP over ATM environment; it provides none of the ARP server functions found in the LS1010. SNMP is used above the IP layer to provide management functionality. This section describes configuring a port on a switch to allow in-band management of the switch CPU.

Configuring In-Band Management in an SVC Environment

This section describes in-band management in an SVC environment. In-band management requires configuring the DSLAM with its own ATM address and that of a single ATM Address Resolution Protocol (ARP) server.

In-band management in an SVC environment is configured by the DSLAM in the following process:

Step 1 The initial IP packet sent by client A triggers a request to the ARP server to look up the IP address and the corresponding ATM address of client B in the ARP server ARP table.

Step 2 The ARP server sends back a response to client A with the matching ATM address.

Step 3 Client A uses the ATM address it just obtained from the ARP server to set up an SVC directly to client B.

Step 4 When client B replies with an IP packet to client A, it also triggers a query to the ARP server.

Note When client B receives the ATM address for client A, it usually discovers it already has a call set up to client A's ATM address and does not set up another call.

After the connection is known to both clients, they communicate directly over the SVC.

The ATM ARP client (the DSLAM) tries to maintain a connection to the ATM ARP server. The ATM ARP server can remove the connection, but the client attempts once each minute to bring the connection back up. No error messages are generated for a failed connection, but the client does not route packets until the ATM ARP server is connected and translates IP network addresses.

For each packet with an unknown IP address, the client (the DSLAM) sends an ATM ARP request to the ARP server. Until that address is resolved, any IP packet routed to the ATM interface causes the client to send another ATM ARP request.

Configuring ATM ARP

In an SVC environment, configure the ATM ARP mechanism on the interface by performing these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*[*.sub_inter#*]	Select the interface to be configured.
2.	**atm nsap-address** *nsap-address* or **atm esi-address** *esi-address*	Specify the NSAP ATM address of the interface. or Specify the end-system-identifier (ESI) address of the interface.
3.	**ip address** *address mask*	Specify the IP address of the interface.
4.	**atm arp-server nsap** *nsap-address*	Specify the ATM address of the ATM ARP server.
5.	**exit**	Exit interface configuration mode.
6.	**atm route** {*addr-prefix*[1]} **atm 0/0 internal**	Configure a static route through the switch to the CPU interface. See the note.

1. First 19 bytes of the NSAP address.

Note You need to specify only a static route when configuring an ARP client using a network service access point (NSAP) address.

NSAP Address Example

This example shows how to configure CPU interface 0/0 of client A using the NSAP address:

```
Client A(config)# interface atm 0/0
Client A(config-if)# $dress 47.0091.8100.0000.1111.1111.1111.1111.1111.1111.00
Client A(config-if)# ip address 123.233.45.1 255.255.255.0
Client A(config-if)# $dress 47.0091.8100.0000.1111.1111.1111.2222.2222.2222.00
Client A(config-if)# exit
Client A(config)# $0.0000.1111.1111.1111.1111.1111.1111 atm 0/0 internal
```

These commands:

1. Identify CPU interface 0/0 for configuration.
2. Configure the interface as an ATM ARP client with NSAP address 47.0091.8100.0000.1111.1111.1111.1111.1111.1111.00.
3. Configure the IP address as 123.322.45.1 with a subnet mask of 255.255.255.0.

4. Configure the ARP server NSAP address as 47.0091.8100.0000.1111.1111.1111.2222.2222.2222.00.
5. Exit interface configuration mode.
6. Configure an internal static route with an NSAP address of 47.0091.8100.0000.1111.1111.1111.1111.1111.1111.00 to ATM interface 0/0.

Note In the preceding example, some of the commands extended beyond the single line of the screen and the command line shifted ten spaces to the left. The dollar sign ($) indicates this shift.

ESI Example

This example shows how to configure CPU interface 0/0 of client A (Figure 13-1), using the ESI:

```
Client A(config)# interface atm 0/0
Client A(config-if)# atm esi-address 0041.0b0a.1081.40
Client A(config-if)# ip address 123.233.45.1 255.255.255.0
Client A(config-if)# $7.0091.8100.0000.1111.1111.1111.2222.2222.2222.00
Client A(config-if)# exit
```

These commands:

1. Identify CPU interface 0/0 for configuration.
2. Configure the interface as an ATM ARP client with end-system identifier 0041.0b0a.1081.40.
3. Configure the interface IP address as 123.233.45.1 with a subnet mask of 255.255.255.0.
4. Specify the ARP server NSAP address as 47.0091.8100.0000.1111.1111.1111.2222.2222.2222.00

Note In the preceding example, one command extended beyond the single line of the screen and the command line shifted ten spaces to the left. The dollar sign ($) indicates this shift.

Show ATM ARP Example

In this example, the **show atm arp** command displays the configuration of the switch interface 0/0:

```
Switch# show atm arp

Note that a '*' next to an IP address indicates an active call

      IP Address        TTL     ATM Address
ATM0/0:
    * 10.0.0.5          19:21   47009181005670000000000112200410b0a108140
```

Show ATM MAP Example

This example displays the map-list configuration of the switch static map and IP-over-ATM interfaces:

```
Switch# show atm map
Map list ATM0/0_ATM_ARP : DYNAMIC
arp maps to NSAP 36.0091810000000003D5607900.0003D5607900.00
        , connection up, VPI=0 VCI=73, ATM0/0
ip 5.1.1.98 maps to NSAP 36.0091810000000003D5607900.0003D5607900.00
        , broadcast, connection up, VPI=0 VCI=77, ATM0/0

Map list ip : PERMANENT
ip 5.1.1.99 maps to VPI=0 VCI=200
```

Configuring In-Band Management in a PVC Environment

This section describes how to configure in-band management in a PVC environment. The ATM Inverse ARP mechanism is applicable to networks that use PVCs, where connections are established but the network addresses of the remote ends are not known.

In a PVC environment, configure the ATM Inverse ARP mechanism by performing the tasks:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**ip address** *address mask*	Specify the IP address of the interface.
3.	**atm pvc** *vpi vci* **encap aal5snap [inarp** *minutes*]	Create a PVC and enable Inverse ARP on it.

Repeat these tasks for each PVC you want to create.

The **inarp** *minutes* interval specifies how often Inverse ARP datagrams are sent on this virtual circuit. The default value is 15 minutes.

Note The ATM ARP and Inverse ATM ARP mechanisms work with IP only. All other protocols require **map-list** command entries to operate.

Example

This example configures an IP-over-ATM interface in a PVC environment and displays the map-list configuration of the switch static map and in-band management interfaces.

These commands:

1. Identify CPU interface 0/0 for configuration.
2. Configure the IP address on the interface as 11.11.11.11.
3. Create an ATM PVC with AAL5SNAP encapsulation, inverse ARP set to ten minutes, on ATM interface 0/0 VPI = 50 VCI = 100.
4. Display the in-band interface configuration.

```
DSLAM(config)# interface atm 0/0
DSLAM(config)# ip address 11.11.11.11
DSLAM(config-if)# atm pvc 0 100 encap aal5snap inarp 10 interface atm 0/0 50 100

DSLAM# show atm map
Map list yyy : PERMANENT
ip 1.1.1.2 maps to VPI=0 VCI=200

Map list zzz : PERMANENT

Map list a : PERMANENT

Map list 1 : PERMANENT

Map list ATM0/0_ATM_ARP : DYNAMIC
arp maps to NSAP 47.009181005670000000001122.00410B0A1081.40
        , connection up, VPI=0 VCI=85, ATM0/0
ip 10.0.0.5 maps to NSAP 47.009181005670000000001122.00410B0A1081.40
        , broadcast, ATM0/0
```

Mapping a Protocol Address to a PVC

The ATM interface supports a static mapping scheme that identifies the ATM address of remote hosts or switches. This IP address is specified as a PVC or as an NSAP address for SVC operation. Configuration for both PVC and SVC map lists are described in these sections:

- Configuring a PVC-Based Map List
- Configuring an SVC-Based Map List

Configuring a PVC-Based Map List

This section describes how to map a PVC to an address, which is a required task if you are configuring a PVC.

You can enter mapping commands as groups. To do so, create a map list and then associate the map list with an interface. Begin with these tasks:

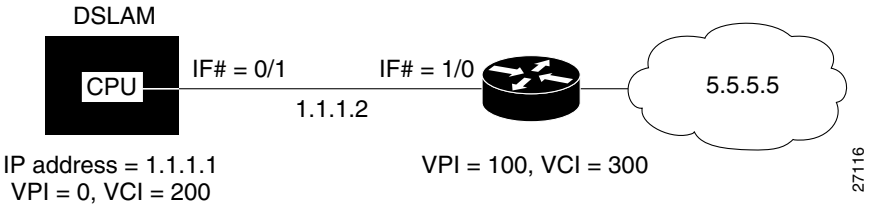
Step	Command	Task
1.	**ip host-routing**	Enable IP host based routing.
2.	**interface atm** *slot/port*[*.sub_inter#*]	Specify an ATM interface and enter interface configuration mode.
3.	**ip** *A.B.C.D mask*	Enter the IP address and subnet mask associated with this interface.
4.	**map-group** *name*	Enter the map group name associated with this PVC.
5.	**atm pvc** *vpi vci* **[encap** *aal5lane aal5mux aal5snap***]** [**upc** *upc*] [**pd** *pd*] [**rx-cttr** *index*] [**tx-cttr** *index*] **interface atm** *slot/port*[*.sub_inter#*] *vpi vci* [**upc** *upc*]	Configure the PVC.
6.	**exit**	Exit interface configuration mode.
7.	**ip route** *A.B.C.D mask* [*A.B.C.D* | **atm** | **ethernet** | **null**]	Configure an IP route to the router.
8.	**map-list** *name*	Create a map list by naming it, and enter map-list configuration mode.
9.	**ip** *A.B.C.D* **atm-nsap** *address* | **atm-vc** *vci* {**aal5mux** *encapsulation* | **broadcast** *pseudo-broadcast* | **class** *class-name*}	Associate a protocol and address to a specific virtual circuit.

You can create multiple map lists, but only one map list can be associated with an interface. Different map lists can be associated with different interfaces.

Example

Figure 13-1 illustrates a connection configured with a PVC map list.

Figure 13-1 PVC Map List Configuration Example



The commands used to configure the connection in Figure 13-1 are

```
DSLAM(config)# ip host-routing
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# ip address 1.1.1.1 255.0.0.0
DSLAM(config-if)# map-group yyy
DSLAM(config-if)# atm pvc 0 200 encap aal5snap interface atm 0/1 100 300
DSLAM(config-if)# exit
DSLAM(config)# ip route 1.1.1.1 255.0.0.0 1.1.1.2
DSLAM(config)# map-list yyy
DSLAM(config-map-list)# ip 1.1.1.2 atm-vc 200
DSLAM(config-map-list)# end
```

These commands enable IP host-based routing.

1. Change to interface configuration mode on ATM CPU interface 0/0.
2. Configure the interface with map group name “yyy.”
3. Configure an internal cross-connect PVC from the CPU interface to ATM interface 0/1 VPI 100 and VCI 300.
4. Exit interface configuration mode.
5. Configure a static IP route between the DSLAM and the router.
6. Change to map list configuration mode and create a map group with the name “yyy.”
7. Associate the map list to the IP network connection 1.1.1.2 and ATM VC 200 configured on ATM interface 0/1.

Example

This example displays the map-list configuration of the DSLAM at interface 0/0:

```
DSLAM# show atm map
Map list yyy : PERMANENT
ip 1.1.1.2 maps to VPI=0 VCI=200
```

Configuring an SVC-Based Map List

This section describes how to map an SVC to an NSAP address. This is a required task if you are configuring an SVC.

You can enter mapping commands as groups. To do so, create a map list and then associate it with the map list interface. Perform these tasks:

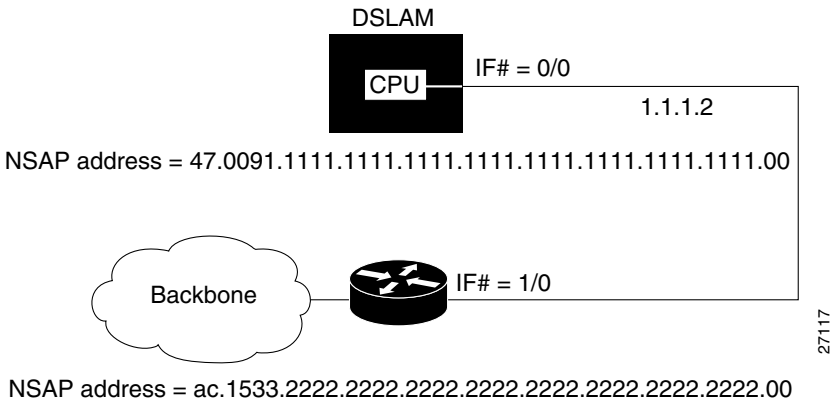
Step	Command	Task
1.	**ip host-routing**	Enable IP host-based routing.
2.	**interface atm** *slot/port*[*.sub_inter#*]	Specify an ATM interface and enter interface configuration mode.
3.	**ip** *A.B.C.D mask*	Enter the IP address and subnet mask associated with this interface.
4.	**atm nsap-address** *20-octet NSAP address*	Configure the interface NSAP address.
5.	**map-group** *name*	Enter the map-group name associated with this PVC.
6.	**exit**	Exit interface configuration mode.
7.	**map-list** name	Create a map list by naming it, and enter map-list configuration mode.
8.	**ip** *A.B.C.D* **atm-nsap** *address* | **atm-vc** *vci* {**aal5mux** *encapsulation* | **broadcast** *pseudo-broadcast* | **class** *class-name*}	Associate a protocol and address to a specific virtual circuit.

You can create multiple map lists, but only one map list can be associated with an interface. Different map lists can be associated with different interfaces.

Examples

Figure 13-2 illustrates an SVC connection configured with a map list.

Figure 13-2 SVC Map List Configuration Example



This example shows the commands used to configure the connection in Figure 13-2:

```
DSLAM(config)# ip host-routing
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# ip address 1.1.1.1 255.0.0.0
DSLAM(config-if)# map-group zzz
DSLAM(config-if)# atm nsap-address 47.0091.1111.1111.1111.1111.1111.1111.1111.1111.00
DSLAM(config-if)# exit
DSLAM(config)# ip route 1.1.1.1 255.0.0.0 1.1.1.2
DSLAM(config)# map-list zzz
DSLAM(config-map-list)# ip 1.1.1.2 atm-nsap
ac.1533.2222.2222.2222.2222.2222.2222.2222.2222.00
```

```
DSLAM(config-map-list)# end
```

These commands:

1. Enable IP host-based routing.
2. Change to interface configuration mode on ATM CPU interface 0/0.
3. Configure the interface with map group name “zzz.”
4. Configure the interface with IP address 1.1.1.1 and a subnet mask.
5. Configure the interface with NSAP address 47.0091.1111.1111.1111.1111.1111.1111.1111.1111.00.
6. Exit interface configuration mode.
7. Configure a static IP route between interface 1.1.1.1 and 1.1.1.2.
8. Switch to map-list configuration mode to map group name “zzz.”
9. Associate the IP interface 1.1.1.2 with NSAP address ac.1533.2222.2222.2222.2222.2222.2222.2222.2222.2222.00.

Example

This example displays the map-list configuration of the DSLAM at interface 0/0:

```
DSLAM# show atm map

Map list yyy : PERMANENT
ip 1.1.1.1 maps to VPI=0 VCI=200
ip 1.1.1.2 maps to VPI=0 VCI=200

Map list zzz : PERMANENT
```

CHAPTER 14

Configuring ATM Accounting and ATM RMON

This chapter describes the ATM accounting and Remote Monitoring (RMON) features used with Cisco DSLAMs with NI-2, and includes these sections:

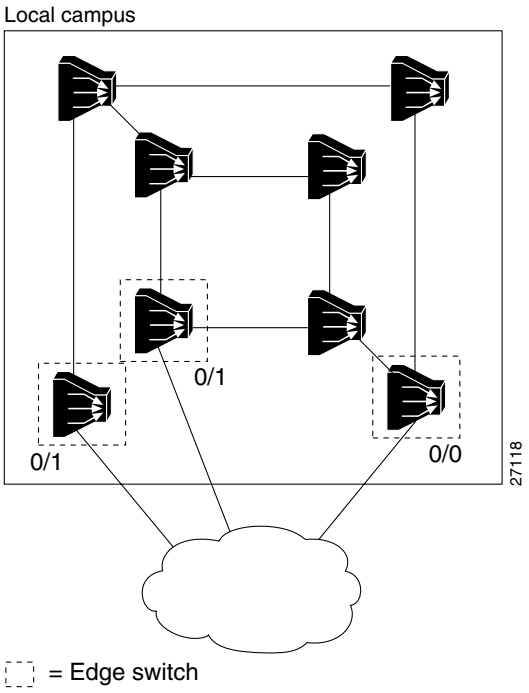
Configuring ATM Accounting

This section describes how to enable and configure the ATM accounting feature in the DSLAM.

ATM Accounting Overview

The ATM accounting feature provides accounting and billing services for virtual circuits (VCs) used on the DSLAM. You enable ATM accounting on an edge switch (or DSLAM) to monitor call setup and traffic activity. A specific interface can be configured to monitor either incoming or outgoing or incoming and outgoing VC use. Figure 14-1 shows a typical ATM accounting environment.

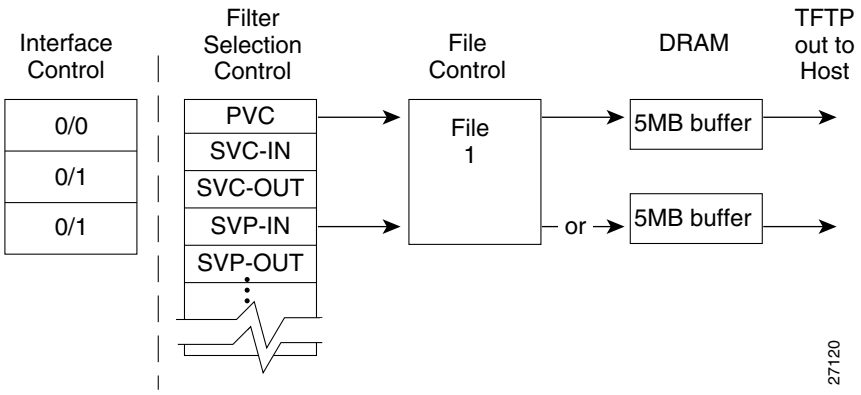
Figure 14-1 ATM Accounting Environment



The edge switches or DSLAMs, connected to the exterior Internet, are connections that require monitoring for accounting and billing purposes.

Switching speeds and number of VCs supported by the DSLAM while monitoring virtual circuit use for accounting purposes can cause the amount of data to be gathered to reach the megabyte range. With such a large amount of data in the ATM accounting files, using traditional Simple Network Management Protocol (SNMP) methods of data retrieval is not feasible. You can store the collected accounting information in a file which you can retrieve using a file transfer protocol. SNMP provides management control of the selection and collection of accounting data. Figure 14-2 shows an interface, filtering, and file configuration example.

Figure 14-2 Interface and File Management for ATM Accounting



A file used for data collection actually corresponds to two memory buffers:

- One buffer is actively saving data.

- The second buffer is passive and ready to have its data either retrieved using Trivial File Transport Protocol (TFTP) or overwritten when the currently active file reaches its maximum capacity.

Using TFTP to download the file is the same process used to download Cisco IOS images and configuration files from the Flash memory to a host.

Configuring Global ATM Accounting

You must enable the ATM accounting feature to start gathering ATM accounting virtual circuit call setup and use data. The ATM accounting feature runs in the background and captures configured accounting data for VC changes such as calling party, called party, or start time and connection type information for specific interfaces to a file.

Caution Enabling ATM accounting could slow the basic operation of the DSLAM.

Note Even if you disable ATM accounting globally, other ATM accounting commands, both global and for individual interfaces, remain in the configuration file.

Use the following commands configure ATM accounting.

Command	Task
atm accounting enable	In global configuration mode, enable ATM accounting for the DSLAM.
show running-config	In privileged EXEC mode, display the ATM accounting status.

Enabling ATM Accounting on an Interface

After you enable ATM accounting, you must configure specific ingress or egress interfaces, usually on edge switches or DSLAMs connected to the external network, to start gathering the ATM accounting data.

To enable ATM accounting on a specific interface, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm accounting**	Enable ATM accounting on the selected interface.

Example

This example shows how to enable ATM accounting on ATM interface 20/0 and displays the result:

```
DSLAM(config)# interface atm 20/0
DSLAM(config-if)# atm accounting
DSLAM# show running-config
Building configuration...
```

```
Current configuration:
!

<Information Deleted>

!
interface ATM20/0
 no keepalive
 atm accounting
!
--More--
!
<information deleted>
```

Configuring the ATM Accounting Selection Table

The ATM accounting selection table determines the connection data to be gathered from the DSLAM. To configure the ATM accounting selection entries, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**atm accounting selection 1**	Configure the ATM accounting selection index number and change to accounting selection mode.
2.	**default** [**connection-type** | **list**]	Reset the ATM accounting selection table configuration to the default.
3.	**connection-types** [**pvc** | **pvp** | **spvc-originator** | **spvc-target** | **spvp-originator** | **spvp-target** | **svc-in** | **svc-out** | **svp-in** | **svp-out**]	Configure the accounting connection types.
4.	**list** *hexadecimal_number*	Configure the list of ATM accounting MIB objects to collect[1].

1. The MIB objects are listed in the *ATM Accounting Information MIB* publication.

The **atm accounting selection** command creates or modifies an entry in the selection table by specifying the fields of the entry.

Note A default selection entry is automatically configured during initial startup and cannot be deleted.

Some features of the ATM selection table configuration include

- An entry in the selection table points to a data collection file.
- A selection entry cannot be deleted when data collection is active.
- A selection entry can point to a nonexistent file, in which case the entry is considered inactive.
- One selection entry can apply to more than one type of VC (for example, SVC and PVC).
- If you modify a selection entry list, the new value is used the next time the data collection cycle begins (for example, the next time the ATM accounting collection file swap occurs).

Note These ATM accounting MIB objects are not supported:

— atmAcctngTransmittedClp0Cells (object number 16)
— atmAcctngReceivedClp0Cells (object number 18)
— atmAcctngCallingPartySubAddress (object number 31)
— atmAcctngCalledPartySubAddress (object number 32)
— atmAcctngRecordCrc16 (object number 33)

Examples

Example	Task
`DSLAM(config)# atm accounting selection 1` `DSLAM(config-acct-sel)# connection-types spvc-originator`	Change to ATM accounting selection configuration mode and add the SVPC originator connection type entry to selection entry 1.
`DSLAM(config)# atm accounting selection 1` `DSLAM(config-acct-sel)# default connection-types`	Change to ATM accounting selection configuration mode and reset the connection types for selection entry 1.
`DSLAM(config)# atm accounting selection 1` `DSLAM(config-acct-sel)# default list`	Change to ATM accounting selection configuration mode and configure the selection list to include all objects.

Example	Task
DSLAM(config)# **atm accounting selection 1** DSLAM(config-acct-sel)# **list 00001000**	Change to ATM accounting selection configuration mode and configure the selection list to include object number 20 (atmAcctngTransmitTrafficDescriptorParam1).

DSLAM# **show atm accounting**
 	
ATM Accounting Info: AdminStatus - UP;
OperStatus : UP
Trap Threshold - 90 percent (4500000 bytes)
Interfaces:
File Entry 1: Name acctng_file1
  Descr: atm accounting data
  Min-age (seconds): 3600
  Failed_attempt : C0
  Sizes: Active 69 bytes (#records 0);
Ready 73 bytes (#records 0)
selection Entry -
  Selection entry 1, subtree -
1.3.6.1.4.1.9.10.18.1.1
  Selection entry 1, list - 00.00.10.00
  Selection entry 1, connType - F0.00
Active selection -
  Selection entry 1, subtree -
1.3.6.1.4.1.9.10.18.1.1
  Selection entry 1, list - FF.FE.BF.FC
  Selection entry 1, connType - F0.00
Debug output
Sig API: Err - 0
New_Conn: OK - 0; Err - 0
Rel_Conn: OK - 0; Err - 0
New_Leg: OK - 0; Err - 0
Rel_Leg: OK - 0; Err - 0
New_Party: OK - 0; Err - 0
Rel_Party: OK - 0; Err - 0
	Shows the ATM accounting status using the **show atm accounting** EXEC command.

Configuring ATM Accounting Files

The ATM accounting data being gathered from the configured selection control table should be directed to a specific ATM accounting file. To configure the ATM accounting files and change to ATM accounting file configuration mode, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**atm accounting file** *filename*	Configure the ATM accounting file and change to accounting file configuration mode.
2.	**collection-modes** {**on-release** | **periodic**}	Configure the time to write to the accounting file.
3.	**default** [**min-age**]	Reset the ATM accounting file configuration to the default.
4.	**description** *64_characters*	Configure a short description for the ATM accounting file.
5.	**enable**	Enable ATM accounting for a specific file.

Step	Command	Task
6.	**failed-attempts {none | regular | soft}**	Configure whether to record failed connection attempts.
7.	**interval** {*60-86400*}	Configure the interval for periodic collection, in seconds.
8.	**min-age** *minutes*	Configure the ATM accounting file minimum age of the VC.

Note You can configure only one ATM accounting file, and you cannot delete that file.

Examples

Example	Task
`DSLAM(config)# atm accounting file acctng_file1` `DSLAM(config-acct-file)# collection-mode on-release`	Enable ATM accounting file configuration mode for acctng_file1 and reconfigure the collection mode on release of a connection.
`DSLAM(config)# atm accounting file acctng_file1` `DSLAM(config-acct-file)# default min-age`	Enable ATM accounting file configuration mode for acctng_file1 and reconfigure the minimum age to the default value.
`DSLAM(config)# atm accounting file acctng_file1` `DSLAM(config-acct-file)# description Main accounting file for engineering`	Enable ATM accounting file configuration mode for acctng_file1 and configure a short description to be displayed in the **show atm accounting file** display and the file header.
`DSLAM(config)# atm accounting file acctng_file1` `DSLAM(config-acct-file)# enable`	Enable ATM accounting file configuration mode for acctng_file1.

Example	Task
DSLAM(config)# **atm accounting file acctng_file1** DSLAM(config-acct-file)# **interval 3600**	Enable ATM accounting file configuration mode for acctng_file1 to collect connection data every hour.
DSLAM# **show atm accounting** ATM Accounting Info: AdminStatus - UP; OperStatus : UP Trap Threshold - 90 percent (4500000 bytes) Interfaces: File Entry 1: Name acctng_file1 Descr: atm accounting data Min-age (seconds): 3600 Failed_attempt : C0 Sizes: Active 69 bytes (#records 0); Ready 73 bytes (#records 0) selection Entry - Selection entry 1, subtree - 1.3.6.1.4.1.9.10.18.1.1 Selection entry 1, list - FF.FE.BF.FC Selection entry 1, connType - F0.00 Active selection - Selection entry 1, subtree - 1.3.6.1.4.1.9.10.18.1.1 Selection entry 1, list - FF.FE.BF.FC Selection entry 1, connType - F0.00 Debug output Sig API: Err - 0 New_Conn: OK - 0; Err - 0 Rel_Conn: OK - 0; Err - 0 New_Leg: OK - 0; Err - 0 Rel_Leg: OK - 0; Err - 0 New_Party: OK - 0; Err - 0 Rel_Party: OK - 0; Err - 0	This example shows the ATM accounting file status using the **show atm accounting** privileged EXEC command.

Controlling ATM Accounting Data Collection

To configure ATM accounting collection, use this command in privileged EXEC mode:

Command	Task
atm accounting collection {collect-now | swap} *filename*	Configure the ATM accounting data collection.

Examples

Example	Task
DSLAM# **atm accounting collection collect-now accntg_file1**	Collect into the specified file all VCs older than the minimum age VCs and displays the result.

DSLAM# **atm accounting collection swap acctng_file1**
DSLAM# **show atm accounting**
 	
ATM Accounting Info: AdminStatus - UP; OperStatus : DOWN
Trap Threshold - 90 percent (4500000 bytes)
Interfaces:
File Entry 1: Name acctng_file1
  Descr: atm accounting data
  Min-age (seconds): 3600
  Failed_attempt : C0
No file buffers initialized
selection Entry -
  Selection entry 1, subtree - 1.3.6.1.4.1.9.10.18.1.1
  Selection entry 1, list - FF.FE.BF.FC
  Selection entry 1, connType - F0.00
Active selection -
  Selection entry 1, subtree - 1.3.6.1.4.1.9.10.18.1.1
  Selection entry 1, list - FF.FE.BF.FC
  Selection entry 1, connType - F0.00
 	
Debug output
Sig API: Err - 0
New_Conn: OK - 0; Err - 0
Rel_Conn: OK - 0; Err - 0
New_Leg: OK - 0; Err - 0
Rel_Leg: OK - 0; Err - 0
New_Party: OK - 0; Err - 0
Rel_Party: OK - 0; Err - 0
	Swap the buffers corresponding to the file named **acctng_file1**.

Configuring ATM Accounting SNMP Traps

This section describes configuring the SNMP server and traps for ATM accounting.

Using an SNMP network management system to monitor the status of the ATM accounting file being created requires two tasks:

- Configuring ATM Accounting Trap Generation
- Configuring SNMP Server for ATM Accounting

Configuring ATM Accounting Trap Generation

To configure ATM accounting SNMP traps, use this command in global configuration mode:

Command	Task
atm accounting trap threshold *percent-value*	Configure the ATM accounting file size threshold to generate an SNMP trap.

Example

This example configures ATM accounting SNMP traps to be sent when the file size reaches 85 percent full and displays the result:

```
DSLAM(config)# atm accounting trap threshold 85
DSLAM# show atm accounting
ATM Accounting Info:     AdminStatus - UP;      OperStatus : UP
Trap Threshold - 90 percent (4500000 bytes)
Interfaces:
File Entry 1: Name acctng_file1
    Descr: atm accounting data
    Min-age (seconds): 3600
    Failed_attempt : C0
    Sizes: Active 69 bytes (#records 0); Ready 73 bytes (#records 0)
selection Entry -
    Selection entry 1, subtree - 1.3.6.1.4.1.9.10.18.1.1
    Selection entry 1, list - FF.FE.BF.FC
    Selection entry 1, connType - F0.00
Active selection -
    Selection entry 1, subtree - 1.3.6.1.4.1.9.10.18.1.1
    Selection entry 1, list - FF.FE.BF.FC
    Selection entry 1, connType - F0.00

Debug output
Sig API: Err - 0
New_Conn: OK - 0; Err - 0
Rel_Conn: OK - 0; Err - 0
New_Leg: OK - 0; Err - 0
Rel_Leg: OK - 0; Err - 0
New_Party: OK - 0; Err - 0
Rel_Party: OK - 0; Err - 0
```

Configuring SNMP Server for ATM Accounting

This section describes configuring the SNMP server ATM accounting traps.

To configure SNMP ATM accounting traps, perform these tasks in global configuration mode:

Step	Command	Task
1.	**snmp-server enable traps atm-accounting**	Enable SNMP server ATM accounting trap generation.
2.	**snmp-server host** *ip_address* [*community-string*] **atm-accounting**	Configure the ATM accounting file size threshold to generate an SNMP trap.

Example

This example shows how to enable SNMP server ATM accounting traps and configure the SNMP server host at IP address 1.2.3.4 with community string *public* for ATM accounting:

```
DSLAM(config)# snmp-server enable traps atm-accounting
```

```
DSLAM(config)# snmp-server host 1.2.3.4 public atm-accounting
```

Displaying SNMP Server ATM Accounting Configuration

To display the SNMP server ATM accounting configuration, use this privileged EXEC command:

Command	Task
show running-config	Display the SNMP server ATM accounting configuration.

Example

This example shows the SNMP server ATM accounting configuration using the **show running-config** privileged EXEC command:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!
version XX.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
!
username dplatz
ip rcmd rcp-enable
ip rcmd remote-host dplatz 171.69.194.9 dplatz
ip rcmd remote-username dplatz
atm template-alias byte_wise 47.9*f8.33...
atm template-alias bit_set 47.9f9(1*0*)88ab...
atm template-alias training 47.1328...
atm accounting enable
atm accounting trap threshold 85
!
<Information Deleted>

no ip classless
atm route 47.0091.8100.0000.0000.0ca7.ce01... ATM3/0/0
snmp-server enable traps chassis-fail
snmp-server enable traps chassis-change
snmp-server enable traps atm-accounting
snmp-server host 1.2.3.4 public atm-accounting
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Using TFTP to Copy the ATM Accounting File

After the ATM accounting file is written to DRAM, you must configure TFTP to allow network requests to copy the accounting information to a host for processing. To do this, use this command in global configuration mode:

Command	Task
tftp-server atm-accounting *filename ip_access_num*	Use TFTP to copy the ATM accounting file to an IP host.

Example

This example allows the TFTP service to copy the ATM accounting file *acctng_file1* to the IP access belonging to requesting host number 1:

```
DSLAM(config)# tftp-server atm-accounting acctng_file1 1
```

Configuring ATM RMON

This section describes the process you use to configure ATM RMON on the DSLAM.

RMON Overview

The ATM RMON feature allows you to monitor network traffic for fault monitoring or capacity planning. The ATM RMON feature is an extension of an existing, well-known RMON standard and provides high-level per-host and per-conversation statistics in a standards-track MIB similar to these RMON MIBs:

- RMON-1 MIB—RFC 1757
- RMON-2 MIB—RFC 2021 and 2074

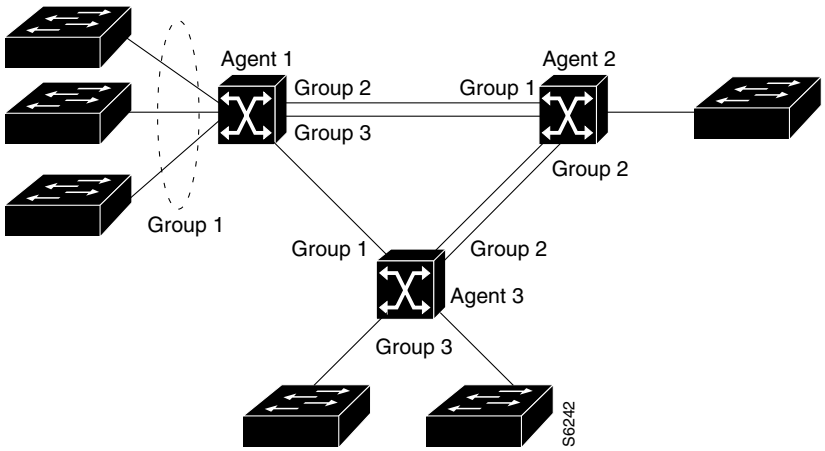
The ATM-RMON counter uses the per-VC counters already maintained in the hardware and polled by the software. The ATM RMON agent can report cell traffic statistics by monitoring connection management activity. At connection setup and release time, some ATM-RMON bookkeeping code executes. The amount of information varies, depending on the ATM RMON configuration. The ATM-RMON bookkeeping capability significantly reduces the CPU requirements for ATM-RMON, and allows collecting statistics on several or all of the DSLAM ports at one time.

The ATM-RMON agent uses the 64-bit version of each cell counter if 64-bit counter support is present in the SNMP master-agent library.

Configuring Port Select Groups

RMON used to allow the collection of connection information on a per-interface basis only. ATM RMON allows a group of ports to be configured as an aggregate. The port select group defines this *collection unit* used by the ATM RMON agent to gather host and matrix connection data. For example, in Figure 14-3, agent 1 includes a port selection group 1 made up of ports.

Figure 14-3 ATM RMON Port Select Group Examples



Before any data collection can begin, you must define an active port select group. To configure and access port select group structures, you can use the command-line interface (CLI) and SNMP modules.

To configure an RMON port selection group, use this command in global configuration mode:

Command	Task
atm rmon portselgrp *number* {**descr** *label* | **host-prio** *value_1-3* | **host-scope** *value_1-3* | **matrix-prio** *value_1-3* | **matrix-scope** *value_1-3* | **maxhost** *number* | **maxmatrix** *number* | **nostats** [*label* | *value_1-3* | *value_1-3* | *value_1-3* | *value_1-3* | *number* | *number* | **owner**] | **owner** *name*}	Configure the ATM RMON port selection group.

Example

This example configures port selection group 7 with these values and displays the result:

- Maximum host count of 500
- Maximum matrix count of 2000
- Host priority of 1
- Owner name “nms 3”

```
DSLAM(config)# atm rmon portselgrp 7 maxhost 500 maxmatrix 2000 host-prio 1 owner “nms 3”
DSLAM# show atm rmon stats 3
PortSelGrp: 3   Collection: Enabled     Drops: 0
  CBR/VBR: calls: 0/0  cells: 0 connTime: 0 days 00:00:00
  ABR/UBR: calls: 0/0  cells: 0 connTime: 0 days 00:00:00
```

Adding Interfaces to a Port Select Group

Before the port selection group can begin gathering host and matrix connection information, you must add an interface or group of interfaces to the port selection group.

To add an interface to an ATM RMON port selection group, use these commands, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm rmon collect** *port_sel_group*	Configure the interface to an ATM RMON port selection group.

Examples

Example	Task

`DSLAM(config)# interface atm 0/0`
`DSLAM(config-if)# atm rmon collect 6`
`DSLAM# show atm rmon host 6`
`PortSelGrp: 6  Collection: Enabled`
`Drops: 0`
	Add ATM interface 0/0 to ATM RMON port selection group 6 and displays the result.

`DSLAM# show atm rmon matrix 6`
`PortSelGrp: 6  Collection: Enabled`
`Drops: 0`
	Display the ATM RMON matrix configuration for port selection group 6 using the **show atm rmon matrix** command from user EXEC mode.

`DSLAM# show atm rmon stats 6`
`PortSelGrp: 6  Collection: Enabled`
`Drops: 0`
`  CBR/VBR: calls: 0/0  cells: 0`
`connTime: 0 days 00:00:00`
`  ABR/UBR: calls: 0/0  cells: 0`
`connTime: 0 days 00:00:00`
	Display the ATM RMON statistics configuration for port selection group 6 using the **show atm rmon stats** command from user EXEC mode.

`DSLAM# show atm rmon status`
`PortSelGrp: 1 Status: Enabled`
`Hosts:  4/no-max  Matrix:  4/no-max`
`  ATM0/0  ATM0/1`
`PortSelGrp: 2 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
`  ATM0/2`
`PortSelGrp: 3 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
`  ATM1/1 ATM1/2`
`PortSelGrp: 4 Status: Enabled`
`Hosts:  0/1  Matrix:  0/5`
`  ATM0/1`
`PortSelGrp: 5 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
`  ATM0/2`
`PortSelGrp: 6 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
`  ATM0/0`
`PortSelGrp: 7 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
`  ATM0/0`
`PortSelGrp: 8 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
 	
`PortSelGrp: 9 Status: Enabled`
`Hosts:  0/no-max  Matrix:  0/no-max`
	Display the ATM RMON status for all port selection groups using the **show atm rmon status** command from user EXEC mode.

Enabling Data Collection

Use the **atm rmon enable** command to start ATM RMON data collection.

Note If you disable ATM RMON the configuration remains but becomes inactive (similar to using the **shutdown** command on an interface).

To enable ATM RMON data collection, use this command in global configuration mode:

Command	Task
atm rmon enable	Enable ATM RMON.

Example

This example shows the ATM RMON configuration using the **show running-config** privileged EXEC command:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!

<information deleted>

ip default-gateway 172.20.53.206
no ip classless
snmp-server community public RW
snmp-server location racka-cs:2016
snmp-server contact abierman
atm rmon portselgrp 1 host-scope 3 matrix-scope 3
atm rmon portselgrp 2 host-scope 3 matrix-scope 3 descr "router port 2" owner
rubble"
atm rmon portselgrp 3 host-scope 3 matrix-scope 3 descr "test" owner "bam_bam"
atm rmon portselgrp 4 maxhost 1 maxmatrix 5 host-scope 1 descr "no active ports" owner "wilma"
atm rmon portselgrp 5
atm rmon portselgrp 6 matrix-prio 1
atm rmon portselgrp 7 host-scope 3 matrix-scope 3 descr "CPU port" owner "pebbles"
atm rmon portselgrp 8
atm rmon portselgrp 9
atm rmon enable
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 login
!
end
```

Configuring an RMON Event

To configure an RMON event, use this command in global configuration mode:

Command	Task
rmon event *number* {**description** *name* | **log** [**description** *name* | **owner** *name* | **trap** *community_string*] | **owner** *name* | **trap** *community_string*}	Configure an RMON event.

Example

This example shows how to configure a generated RMON event and displays the result:

- Named 1
- Description string test
- Owner “nms 3”
- SNMP trap with the community string test

```
DSLAM(config)# rmon event 1 description test owner nms_3 trap test
DSLAM# show rmon events
Event 1 is active, owned by nms_3
 Description is test
 Event firing causes trap to community test, last fired 00:00:00
```

Configuring an RMON Alarm

You can configure RMON alarm generation if any of the configured parameters are met.

To configure RMON alarms, use this command in global configuration mode:

Command	Task
rmon alarm *number mib_object interval* {**absolute rising-threshold** *value* **falling-threshold** *value* **owner** *name* | **delta rising-threshold** *value* **falling-threshold** *value* **owner** *name*}	Configure the ATM RMON alarm.

Example

This example configures RMON alarm number 1 to generate an alarm and displays the result:

- If the MIB atmHostHCCells exceed 500
- If each sample, in absolute mode, shows:
  - Rising threshold exceeding 10,000
  - Falling threshold falling below 1000
- The RMON alarm number 1 sends the alarm to the owner “nms 3”

```
DSLAM(config)# rmon alarm 1 atmHostInHCCells 500 absolute rising-threshold 10000
falling-threshold 1000 owner "nms 3"
DSLAM# show rmon alarms events
Event 1 is active, owned by nms 3
 Description is test
 Event firing causes trap to community test, last fired 00:00:00
```

```
Alarm table is empty
```

CHAPTER 15

Configuring Signaling Features

This chapter describes the signaling features for Cisco DSLAMs with NI-2. It includes these sections:

Configuring Signaling IE Forwarding

You enable signaling information element (IE) forwarding of the specified IE from the calling party to the called party.

Note The default is to transfer all of the information elements in the signaling message.

To configure an interface signaling IE transfer, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**atm signalling ie forward** {**aal-info** | **all** | **bli-repeat-ind** | **called-subaddress** | **calling-number** | **higher-layer-info** | **lower-layer-info** | **unknown-ie**}	Configure the signaling information element forwarding.

Example

This example disables signaling of all forwarded IEs on ATM interface 0/0 and displays the result:

```
DSLAM(config)# interface atm 0/0
DSLAM(config-if)# no atm signalling ie forward all
DSLAM# show running-config
Building configuration...

Current configuration:
```

```
!
version XX.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM

!

<information deleted>

!
interface ATM0/0
 no atm signallling ie forward calling-number
 no atm signallling ie forward calling-subaddress
 no atm signallling ie forward called-subaddress
 no atm signallling ie forward higher-layer-info
 no atm signallling ie forward lower-layer-info
 no atm signallling ie forward blli-repeat-ind
 no atm signallling ie forward aal-info
!
interface ATM0/1
!
interface ATM0/2
!
```

Configuring E.164 Addresses

E.164 support allows networks that use E.164 ATM address formats (for example, 45.000001234567777F00000000.000000000000.00) to work with networks that use E.164 address formats (for example, 1–123–456–7777). Generally, you can use E.164 ATM addresses in ATM networks, and E.164 addresses in telephone networks.

There are several types of E.164 addresses. The DSLAM supports these E.164 address formats:

- Native E.164—An ASCII address that complies with the International Telecommunications Union (ITU) E.164 specification for international telephone numbers. For example, the number 1–800–555–1212 is encoded as 3138303035353531323132. Native E.164 addresses have these properties:
  - Conform to ITU E.164 specification
  - Contain 7 to 15 digits
  - Decimal digits 0 to 9
  - IA5 number, ASCII, 8-bits, MSB = 0
  - Result equals one digit per byte
  - User-Network Interface (UNI) or Interim Interswitch Signaling Protocol (IISP) support only; Private Network-Network Interface (PNNI) does not support E.164 addresses

  These properties are carried in the called and calling party address IEs, which are part of the signaling packets used to set up a call.
- ARB_AESA—a form of ATM End System Address (AESA) with any arbitrary numbering, for example, 47.111111111111111111111111.111111111111.00.

- E164_ZDSP—An E164_AESA address with all zeros after the embedded E.164 number; for example, 45.000001234567777F00000000.000000000000.00. ZDSP means “Zero Domain Specific Part.”
- E164_AESA—An AESA address with an embedded E.164 number; for example, 45.000007654321111FDDDDDDDD.CCCCCCCCCCCC.00. The “D” and “C” characters in this example represent an end system address.

Note AESA is an ATM Forum term for ATM address.

There are three features you can configure on the DSLAM for E.164 address conversion. The feature you choose depends on the address format you are using. The features are as follows:

- E.164 gateway—Use this feature when addresses are in ARB_AESA format and a call must traverse an E.164 network.
- E.164 address autoconversion—Use this feature when addresses are in E164_ZDSP or E.164_AESA format and a call must traverse an E.164 network.
- E.164 address one-to-one translation table—Use this feature when you want to create an E.164 to AESA address translation table manually. This feature is not recommended for most networks.

Caution Manually creating the E.164 to AESA address translation table is a time consuming and error prone process. Cisco recommends that you use either the E.164 gateway or E.164 autoconversion feature instead of the E.164 one-to-one address translation feature.

Proceed to the appropriate subsection for configuration information.

Configuring E.164 Gateway

If your network uses ARB_AESA, you can configure the E.164 gateway feature. To configure the E.164 gateway feature, you must first configure a static ATM route with an E.164 address. Then configure the E.164 address to use on the interface.

This section describes how to configure the E.164 gateway feature and includes these procedures:

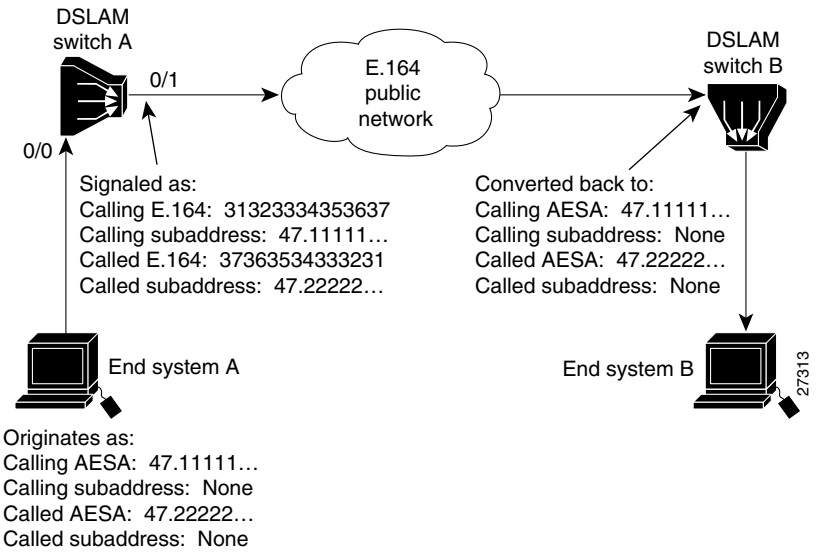
- Configuring a static ATM route with an E.164 address
- Configuring an E.164 address on an interface

When a static route is configured on an interface, all ATM addresses that match the configured address prefix are routed through that interface to an E.164 address.

Signaling uses E.164 addresses in the called and calling party IEs, and uses AESAs in the called and calling party subaddress IEs.

Figure 15-1 illustrates an E.164 gateway configuration.

Figure 15-1 E.164 Gateway Conversion Example



The AESA address is used to initiate the call at the ingress to the public network. The public network routes the call based on the E.164 address. AESA subaddresses are carried through the public network in the subaddress fields. The AESA address is used to complete the call at the egress from the public network.

Note Enter access lists for E.164 addresses in the E164_AESA format, not native E.164 format. For example, if the E.164 address is 7654321, then the E164_AESA format is 45.000000007654321F00000000.000000000000.00. To filter prefix 765, enter the prefix 45.00000000765..., not just 765.... Access lists operate on the called and calling party IEs.

Configuring an E.164 Address Static Route

To configure an E.164 address static route, use this command in global configuration mode:

Command	Task
atm route *atm-address-prefix* **atm** *slot/port* [**e164-address** *e164-address* [**number-type** {**international** | **local** | **national** | **subscriber**}]] [**internal**] [**scope** *1-15*]	At the configure prompt, configure the static route prefix using the E.164 address.

Example

This example uses the **atm route** command to configure a static route using the 13-byte switch prefix 47.00918100000000410B0A1081 to ATM interface 0/0 with the E.164 address 1234567 and displays the result (To complete the E.164 address static route configuration, proceed to the "Configuring an ATM E.164 Address on an Interface" section on page 15-31):

```
DSLAM(config)# atm route 47.00918100000000410B0A1081 atm 0/0 e164-address 7654321
DSLAM# show atm route
Codes: P - installing Protocol (S - Static, P - PNNI, R - Routing control),
       T - Type (I - Internal prefix, E - Exterior prefix, SE -
```

```
                Summary Exterior prefix, SI - Summary Internal prefix,
                ZE - Suppress Summary Exterior, ZI - Suppress Summary Internal)
P  T Node/Port        St Lev Prefix
~ ~~ ~~~~~~~~~~~~~~~ ~~ ~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
S  E 1   ATM0/1     DN 0   47.0091.8100.0000.0001/72
P SI 1   0           UP 0   47.0091.8100.0000.0002.eb1f.fe00/104
R  I 1   ATM0/0     UP 0   47.0091.8100.0000.0002.eb1f.fe00.0002.eb1f.fe00/152
R  I 1   ATM0/0     UP 0   47.0091.8100.0000.0002.eb1f.fe00.4000.0c/128
P SI 1   0           UP 0   47.0091.8100.0000.0040.0b0a.2b81/104
S  E 1   ATM0/0     DN 0   47.0091.8100.0000.0040.0b0a.2b81/104
                              (E164 Address 1234567)
R  I 1   ATM0/0     UP 0   47.0091.8100.0000.0040.0b0a.2b81.0040.0b0a.2b81/152
R  I 1   ATM0/0     UP 0   47.0091.8100.0000.0040.0b0a.2b81.4000.0c/128
```

Configuring an ATM E.164 Address on an Interface

You can configure one E.164 address per ATM port. Signaling uses E.164 addresses in the called and calling party IEs, and uses AESA addresses in the called and calling party subaddress IEs.

To configure an E.164 address on a per-interface basis, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select an interface port.
2.	**atm e164 address** *e164-address*	Associate the E.164 address to the interface.

Example

This example configures the E.164 address 7654321 on ATM interface 0/1 and displays the result:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm e164 address 7654321
DSLAM# show atm interface atm 0/1

Interface:      ATM0/1        Port-type:    oc3suni
IF Status:      UP              Admin Status: up
Auto-config:    enabled         AutoCfgState: completed
IF-Side:        Network         IF-type:      NNI
Uni-type:       not applicable  Uni-version:  not applicable
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.0091.8100.0000.0041.0b0a.1081.4000.0c80.0010.00
ATM E164 Address: 7654321
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      3        0       0       0        0       0           3                3
Logical ports(VP-tunnels):     0
Input cells:    226064          Output cells: 226139
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 147608, Output AAL5 pkts: 147636, AAL5 crc errors: 0
```

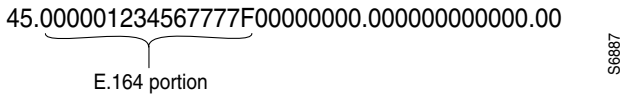
When the E.164 gateway feature is configured, the DSLAM first attempts to make a connection using the E.164 gateway feature. If that connection fails, the DSLAM attempts to make the connection using the E.164 address autoconversion feature. Proceed to the next section for configuration instructions.

Configuring E.164 Address Autoconversion

If your network uses E164_ZDSP or E164_AESA addresses, you can configure E.164 address autoconversion. The E164_ZDSP and E164_AESA addresses include an embedded E.164 number in the E.164 portion of an E.164 ATM address. This embedded E.164 number is used in the autoconversion process.

The E.164 portion of an E.164 ATM address is the first 15 digits following the authority and format identifier (AFI) of 45, shown in Figure 15-2.

Figure 15-2 E.164 Portion of an E.164 ATM Address



The E.164 portion is right-justified and ends with an "F." If all fifteen digits are not being used, the unused digits are filled with zeroes. In Figure 15-2, the embedded E.164 number is 1234567777, but it is signaled at the egress of the DSLAM and in the E.164 public network as 31323334353637373737.

The autoconversion process differs slightly between the E164_ZDSP and E164_AESA address formats. Table 15-1 compares the E.164 address autoconversion process by address type. The main difference between the two types is the way the IEs are signaled at the egress of the DSLAM, as described in the second row of Table 15-1. Note that during the final conversion process, the calling AESA and called AESA return to their original values.

Table 15-1 E164_ZDSP and E164_AESA Address Autoconversion Comparison

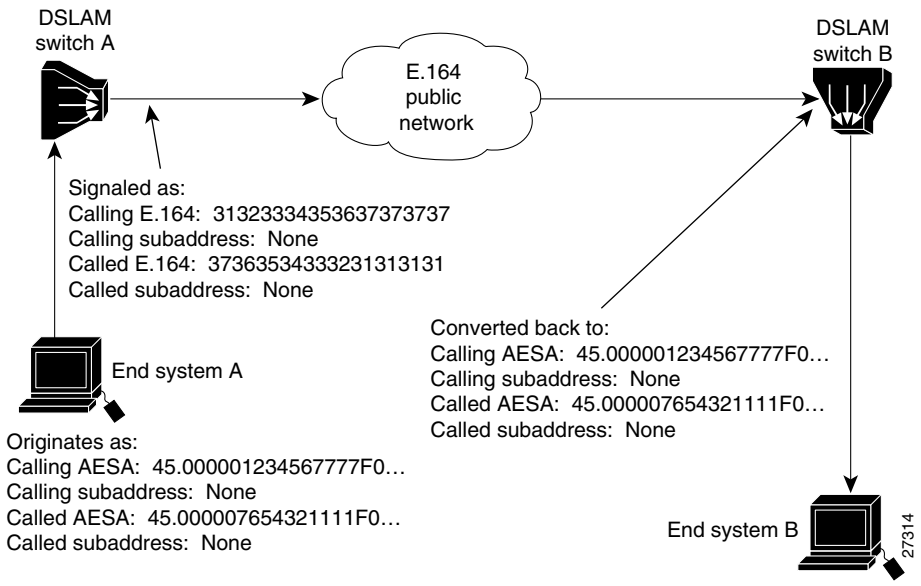
Action	E164_ZDSP	E164_AESA
Originates as	Calling AESA: 45.000001234567777F00000000.000000000000.00 Calling subaddress: None Called AESA: 45.000007654321111F00000000.000000000000.00 Called subaddress: None	Calling AESA: 45.000001234567777FAAAAAAAA.BBBBBBBBBBBB.00 Calling subaddress: None Called AESA: 45.000007654321111FCCCCCCCC.DDDDDDDDDDDD.00 Called subaddress: None

Table 15-1 E164_ZDSP and E164_AESA Address Autoconversion Comparison

Signaled at egress of DSLAM as	Calling E.164: 31323334353637373737 Calling subaddress: None Called E.164: 37363534333231313131 Called subaddress: None	Calling E.164: 31323334353637373737 Calling subaddress: 45.0000012345677777FAAAAAAAA.BBBBBBBBBBBB.00 Called E.164: 37363534333231313131 Called subaddress: 45.000007654321111FCCCCCCCC.DDDDDDDDDDDD.00
Converted back at ingress of DSLAM to	Calling AESA: 45.000001234567777F00000000.000000000000.00 Calling subaddress: None Called AESA: 45.000007654321111F00000000.000000000000.00 Called subaddress: None	Calling AESA: 45.000001234567777FAAAAAAAA.BBBBBBBBBBBB.00 Calling subaddress: None Called AESA: 45.000007654321111FCCCCCCCC.DDDDDDDDDDDD.00 Called subaddress: None

Figure 15-3 shows an example of an E164_ZDSP address autoconversion.

Figure 15-3 E164_ZDSP Sample Address Autoconversion

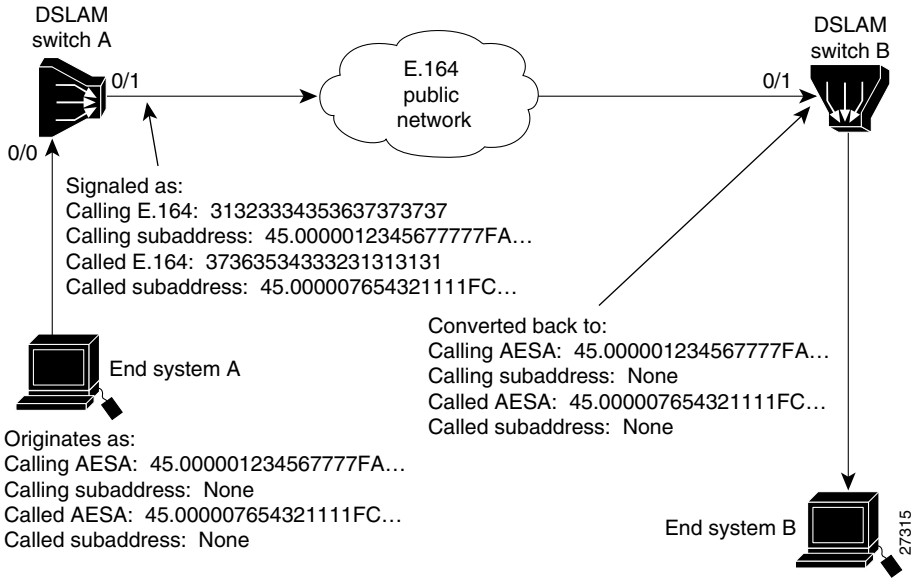


In Figure 15-3, a call (connection) from end system A is placed to end system B on the other side of an E.164 public network. The call originates as an E.164 ATM address and is signaled in native E.164 format at the egress port of DSLAM switch A and within the E.164 public network. When the call reaches the ingress port of DSLAM switch B, at the edge of the E.164 public network, the call is converted back to E.164 ATM address format.

Note The DSLAM routes calls based on the E.164 ATM address (not the native E.164 address).

Figure 15-4 shows an example of an E164_AESA address autoconversion.

Figure 15-4 E164_AESA Address Autoconversion Example



In Figure 15-4, a call from end system A is placed to end system B on the other side of an E.164 public network. The call originates as an E.164 ATM address and at the egress port of DSLAM switch A and within the E.164 public network:

- The E.164 ATM address is signaled in native E.164 format.
- The called party address (45.000007654321111F...) IE is included in the called party subaddress IE.
- The calling party address (45.000001234567777F...) IE is included in the calling party subaddress IE.

When the call reaches the ingress port of DSLAM switch B, at the edge of the E.164 public network, the call is converted back to E.164 ATM address format and:

- The native E.164 address is converted back to an E.164 ATM address.
- The called party subaddress (45.000007654321111F...) IE is returned to the called party address IE.
- The calling party subaddress (45.000001234567777F...) IE is returned to the calling party address IE.

Note Enter access lists for E.164 addresses in the E164_AESA format, not native E.164 format. For example, if the E.164 address is 7654321, then the E164_AESA format is 45.000000007654321F00000000.000000000000.00. To filter prefix 765, enter the prefix 45.00000000765..., not just 765.... Access lists operate on the called and calling party IEs.

E.164 address autoconversion configuration is the same, regardless of which type of address (E164_ZDSP or E164_AESA) your network uses. To configure E.164 address autoconversion, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**atm route** *atm-address-prefix* **atm** *slot/port* [**e164-address** *e164-address* [**number-type** {**international** | **local** | **national** | **subscriber**}]] [**internal**] [**scope** *1-15*]	At the configure prompt, configure the static route prefix with the E.164 address.
2.	**interface atm** *slot/port*	Select the ATM interface.
3.	**atm e164 auto-conversion**	Configure E.164 autoconversion.
4.	**exit**	Return to global configuration mode.

Examples

Command	Task

`DSLAM(config)# atm route 45.000007654321111F atm 0/1`
`DSLAM(config)# int atm 0/1`
`DSLAM(config-if)# atm e164 auto-conversion`
	Configure interface 0/1 of DSLAM switch A in the example networks shown in Figure 15-3 and Figure 15-4.

`DSLAM(config)# atm route 45.000001234567777F atm 0/1`
`DSLAM(config)# int atm 0/1`
`DSLAM(config-if)# atm e164 auto-conversion`
	Configure interface 0/1 of DSLAM switch B.

`DSLAM# show atm interface atm 0/1`
 	
`Interface: ATM0/1 Port-type: oc3suni`
`IF Status: DOWN Admin Status: down`
`Auto-config: disabled AutoCfgState: not applicable`
`IF-Side: Network IF-type: UNI`
`Uni-type: Private Uni-version: V3.0`
`Max-VPI-bits: 8 Max-VCI-bits: 14`
`Max-VP: 255 Max-VC: 16383`
`ConfMaxSvpcVpi: 255 CurrMaxSvpcVpi: 255`
`ConfMaxSvccVpi: 255 CurrMaxSvccVpi: 255`
`ConfMinSvccVci: 33 CurrMinSvccVci: 33`
`Svc Upc Intent: pass Signalling: Enabled`
`ATM Address for Soft VC:`
`47.0091.8100.0000.0002.eb1f.fe00.4000.0c80.0010.00`
`ATM E164 Auto Conversion Interface`
`Configured virtual links:`
`PVCLs SoftVCLs SVCLs TVCLs PVPLs SoftVPLs SVPLs Total-Cfgd Inst-Conns`
`2 0 0 0 0 0 0 2 0`
`Logical ports(VP-tunnels): 0`
`Input cells: 0 Output cells: 0`
`5 minute input rate: 0 bits/sec, 0 cells/sec`
`5 minute output rate: 0 bits/sec, 0 cells/sec`
`Input AAL5 pkts: 0, Output AAL5 pkts: 0, AAL5 crc errors: 0`
	Display the E.164 configuration for ATM interface 0/1.

Configuring E.164 Address One-to-One Translation Table

The ATM interface to a public network commonly uses an E.164 address for ATM signaling, with ARB_AESA addresses carried in the subaddress fields of the message.

Caution Manually mapping AESA addresses to E.164 addresses is a time consuming and error prone process. Cisco recommends that you use either the E.164 gateway or E.164 autoconversion feature instead of the E.164 one-to-one address translation feature.

The one-to-one translation table allows signaling to look up the E.164 addresses and the ARB_AESA addresses in a database, allowing a one-to-one correspondence between ARB_AESA addresses and E.164 addresses.

During egress operation, when a signaling message attempts to establish a call out an interface, the called and calling party addresses are in ARB_AESA format.

If the interface has been configured for E.164 translation, signaling attempts to find a match for the ARB_AESA addresses. If found, the E.164 addresses corresponding to the ARB_AESA addresses are placed into the called and calling party addresses. The original ARB_AESA addresses are also placed into the called and calling party subaddresses.

- During ingress operation, if the interface is configured for E.164 translation, the called and calling party addresses are in E.164 format.
- If the original ARB_AESA-formatted called and calling addresses have been carried in subaddresses, then those addresses are used to forward the call.
- If subaddresses are not present due to the network blocking them, or to the switch at the entry to the E.164 network not using subaddresses, signaling attempts to find a match for the ARB_AESA address in the ATM E.164 translation table.
- If matches are found, the ARB_AESA addresses corresponding to the E.164 addresses in the translation table will be placed into the called and calling party addresses. The call is then forwarded using the ARB_AESA addresses.

To configure a one-to-one E.164 translation table:

Step 1 Configure specific ATM interfaces to connect to E.164 public networks to use the translation table.

Step 2 Configure the translation table.

Step 3 Add entries to the translation table for both the called and calling parties.

To configure E.164 translation on the interface, perform these steps, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select an interface port.
2.	**atm e164 translation**	Configure the ATM E.164 interface.
3.	**exit**	Return to EXEC configuration mode.

Step	Command	Task
4.	**atm e164 translation-table**	Change to E.164 ATM configuration mode.
5.	**e164 address** *address* **nsap-address**[1] *nsap_address*	Configure the E.164 translation table.

1. The NSAP address is the same as the ARB_AESA address.

Examples

This example shows how to configure the ATM interface 0/1 to use the one-to-one E.164 translation table:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm e164 translation
DSLAM(config-if)# exit
DSLAM(config)# atm e164 translation-table
DSLAM(config-atm-e164)# e164 address 1111111 nsap-address 11.111111111111111111111111.112233445566.11
DSLAM(config-atm-e164)# e164 address 2222222 nsap-address 22.222222222222222222222222.112233445566.22
DSLAM(config-atm-e164)# e164 address 3333333 nsap-address 33.333333333333333333333333.112233445566.33
```

These commands:

1. Change to interface configuration mode for ATM interface 0/1.
2. Enable ATM E.164 translation on the interface.
3. Exit interface configuration mode.
4. Change to ATM E.164 configuration mode.
5. Add the E.164 address 1111111 to the ARB_AESA address 11.11111... translation table entry.
6. Add the E.164 address 2222222 to the ARB_AESA address 22.22222... translation table entry.
7. Add the E.164 address 3333333 to the ARB_AESA address 33.33333... translation table entry.

This example shows how to display the E.164 translation table configuration:

```
DSLAM# show running-config
Building configuration...

Current configuration:
!
version XX.X
no service pad
service udp-small-servers
service tcp-small-servers
!
hostname DSLAM
!
!
username dtate
!
atm e164 translation-table
 e164 address 1111111 nsap-address 11.111111111111111111111111.112233445566.11
 e164 address 2222222 nsap-address 22.222222222222222222222222.112233445566.22
 e164 address 3333333 nsap-address 33.333333333333333333333333.112233445566.33
!
atm service-category-limit cbr 64544
atm service-category-limit vbr-rt 64544
atm service-category-limit vbr-nrt 64544
atm service-category-limit ubr 64544
atm address 47.0091.8100.0000.0040.0b0a.2b81.0040.0b0a.2b81.00
```

```
 --More--
<information deleted>
```

This example shows how to display the E.164 configuration for ATM interface 0/1:

```
DSLAM# show atm interface atm 0/1

Interface:      ATM0/1        Port-type:    oc3suni
IF Status:      DOWN            Admin Status: administratively down
Auto-config:    enabled         AutoCfgState: waiting for response from peer
IF-Side:        Network         IF-type:      UNI
Uni-type:       Private         Uni-version:  V3.0
Max-VPI-bits:   8               Max-VCI-bits: 14
Max-VP:         255             Max-VC:       16383
Svc Upc Intent: pass            Signalling:   Enabled
ATM Address for Soft VC: 47.9999.9999.0000.0000.0000.0216.4000.0c80.0010.00
ATM E164 Translation Interface
Configured virtual links:
  PVCLs SoftVCLs   SVCLs   PVPLs SoftVPLs   SVPLs  Total-Cfgd  Installed-Conns
      2        0       0       0        0       0           2                0
Logical ports(VP-tunnels):     0
Input cells:    0               Output cells: 0
5 minute input rate:             0 bits/sec,       0 cells/sec
5 minute output rate:            0 bits/sec,       0 cells/sec
Input AAL5 pkts: 0, Output AAL5 pkts: 0, AAL5 crc errors: 0
```

Configuring Signaling Diagnostics Tables

Use signaling diagnostics to diagnose a specific call failure in your network and pinpoint the location of the call failure along with the reason for the failure.

To do this, you must:

1. Configure a signaling diagnostics table that stores the filtering criteria and a filter index, an integer value between 1 and 50, used to uniquely identify each set of filtering criteria you select. Each filtering criteria occupies one entry in the signaling diagnostics table. Each entry in the filter table is entered using command-line interface (CLI) commands or Simple Network Management Protocol (SNMP).
2. Then the diagnostics software module, when enabled, filters rejected calls based on the entries in your filter table.
3. A successful match in the filter table causes the rejected call information to be stored for analysis.

Note Signaling diagnostics is a tool for troubleshooting failed calls and should not be enabled during normal DSLAM operation.

To configure the signaling diagnostics table entries, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**atm signalling diagnostics enable**	Enable ATM signaling diagnostics
2.	**atm signalling diagnostics** *index*	Change to ATM signaling diagnostics configuration mode.

Step	Command (continued)	Task
3.	**age-timer** *seconds*	Configure the timeout value for the entry, in seconds.
4.	**calling-address-mask** *nsap_address_mask*[2]	Configure a filtering criteria based on the calling address mask value to be used to identify the valid bits of the calling NSAP address of the rejected call.
5.	**called-nsap-address** *nsap_addrress*	Configure a filtering criteria based on the called NSAP address of the rejected call.
6.	**called-address-mask** *nsap_address_mask*[1]	Configure a filtering criteria based on the called address mask value used to identify the valid bits of the calling NSAP address of the rejected call.
7.	**calling-nsap-address** *nsap_address*	Configure a filtering criteria based on the calling NSAP address of the rejected call.
8.	**cast-type** {**p2p** | **p2mp** | **all**}	Configure a filtering criteria based on the cast type of the rejected call. (The default is **all**.)
9.	**clear-cause** *number*[2]	Configure a filtering criteria based on the cleared cause code of the rejected call.
10.	**connection-category** {**soft-vc** | **soft-vp** | **reg-vc** | **all**}	Configure a filtering criteria based on the VC connection category of the rejected call.
11.	**incoming-port atm** *slot/port*	Configure a filtering criteria based on the incoming port of the rejected call.
12.	**max-records** *number*	Configure the maximum number of entries to be stored in the display table for each of the entries in the filter table.
13.	**outgoing-port atm** *slot/port*	Configure a filtering criteria based on the outgoing port of the rejected call.
14.	**purge**	Purge all the filtered records in the filter table.
15.	**scope** {**internal** | **external**}	Configure a filtering criteria based on the scope of the rejected call which either failed internally in the DSLAM or externally on other DSLAMs or switches.
16.	**service-category** {**cbr** | **vbr-rt** | **vbr-nrt** | **ubr** | **all**}	Configure a filtering criteria based on the service category of the rejected call.
17.	**status** [**active** *filter_criteria* | **inactive** *filter_criteria* | **delete** *filter_criteria*]	Configure the status of the entry in the filter table.

1. The combination of the configured *calling_addr_mask* (*called_address_mask*) and the configured *calling_nsap_address* (*called_nsap_address*) are used to filter the rejected call.
2. You can obtain the cause code values from the ATM forum UNI 3.1 specification.

The display table contains the records that were collected based on every filtering criteria in the filter table. Each filtering criteria has only a specified number of records that are stored in the table. After that specified number of records is exceeded, the table is overwritten.

Examples

Example	Task
`DSLAM(config)# atm signalling diagnostics enable`	Enable signaling diagnostics on the DSLAM.

`DSLAM(config)# atm signalling diagnostics 1`
`DSLAM(cfg-atmsig-diag)# service-category cbr`
`DSLAM(cfg-atmsig-diag)# service-category ubr`
`DSLAM(cfg-atmsig-diag)# service-category ubr`
	Configure filter criteria in signaling diagnostics index 1 for call failures based on the service category.

`DSLAM(config)# atm signalling diagnostics 1`
`DSLAM(cfg-atmsig-diag)# connection-category soft-vc`
`DSLAM(cfg-atmsig-diag)# connection-category soft-vc soft-vp`
	Configure filter criteria for call failures based on the category of the virtual circuit.

`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# cast-type p2p p2mp`	Configure filter criteria for calls rejected based on the connection type.
`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# scope internal`	Configure filter criteria for calls that failed internally in the DSLAM.
`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# incoming-port ATM0/2`	Configure the filter entry for filtering failed calls that came in through interface ATM0/2.
`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# outgoing-port ATM0/2`	Configure the filter entry for filtering failed calls that went out through interface ATM0/2.
`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# clearcause 3`	Configure the filter entry for filtering failed calls based on the clear cause value 3 (destination unreachable).

`DSLAM(config)# atm signalling diagnostics 1`
`DSLAM(cfg-atmsig-diag)# calling-nsap-address`
`47.009181000000061705BD901.010203040506.0.`
	Configure filter criteria for calls rejected based on the calling NSAP address of the call.

`DSLAM(config)# atm signalling diagnostics 1`
`DSLAM(cfg-atmsig-diag)# called-nsap-address`
`47.009181000000061705BD901.010203040506.0`
	Configure filter criteria for calls rejected based on the called NSAP address of the call.

`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# called-address-mask ff.ff.ff.00`	Configure filter criteria for calls rejected based on the called address mask of the call.
`DSLAM(config)# atm signalling diagnostics 1` `DSLAM(cfg-atmsig-diag)# calling-address-mask ff.ff.ff.00`	Configure filter criteria for calls rejected based on the calling address mask of the call.

Example (continued)	Task
DSLAM(config)# **atm signalling diagnostics 1** DSLAM(cfg-atmsig-diag)# **age-timer 3600**	This example shows how to specify the timeout value for the entry in seconds:
DSLAM(config)# **atm signalling diagnostics 1** DSLAM(cfg-atmsig-diag)# **purge**	Purge all the filtered records corresponding to this entry in the filter table.
DSLAM(config)# **atm signalling diagnostics 1** DSLAM(cfg-atmsig-diag)# status delete	Delete an index entry in the filter table.
DSLAM(config)# **atm signalling diagnostics 1** DSLAM(cfg-atmsig-diag)# **max-records 40**	Specify the maximum number of entries to be stored in the display table for each of the entries in the filter table
DSLAM# **show atm signalling diagnostics record 1** D I S P L A Y I N D E X 1 --- Scope: internal, Cast Type: p2p, Conn Indicator: Setup Failure Connection Kind: switched-vc Service Category: UBR (Unspecified Bit Rate) Clear Cause: 0x29, Diagnostics: NULL Incoming Port: ATM0/2, Outgoing Port:ATM0/1 Calling-Address: 47.009181000000006011000000.470803040506.00 Calling-SubAddr: NULL Called-Address : 47.00918100000006083C42C01.750203040506.00 Called-SubAddr : NULL Crankback Type : No Crankback DTL's : NodeId:56:160:47.009181000000006011000000.006083AB9001.00 Port: 0/1:2 NodeId:56:160:47.0091810000000603E7B4101.00603E7B4101.00 Port: 0/0:2 NodeId:56:160:47.00918100000006083C42C01.006083C42C01.00 Port: 0	Display the signaling diagnostic records for index 1.

Example (continued)	Task
DSLAM# **show atm signalling diagnostics filter 1** F I L T E R I N D E X 1 ---------- Scope: internal, Cast Type: p2mp Connection Kind: soft-vc Service Category: CBR (Constant Bit Rate) UBR (Unspecified Bit Rate) Clear Cause: 0, Initial TimerValue: 600 Max Records: 20, NumMatches: 0, Timer expiry: 600 Incoming Port: ATM0/1, Outgoing Port: ATM0/2 Calling Nsap Address:47.111122223333444455556666.777788889999.00 Calling Address Mask:FF.FFFFFF00000000000000000.000000000000.00 Called Nsap Address :47.111122223333444455556666.777788889999.01 Called Address Mask :FF.FFFFFF00000000000000000.000000000000.00 Status : active	Display the signaling diagnostics data for filter index 1.
DSLAM# **show atm signalling diagnostics status** Signaling diagnostics disabled globally	Display the signaling diagnostics status.

Configuring Closed User Group Signaling Overview

You can configure a closed user group (CUG) to form restricted access groups (virtual private networks). You can define different CUGs and a specific user can be a member of one or more CUGs. Members of a CUG can communicate among themselves, but not with users outside the group. Specific users can have additional restrictions that prevent them from originating or receiving calls from other members of the CUG. You can also specify additional restrictions on originating and receiving calls to or from members of other CUGs.

For example, if you configure three CUGs (A, B, and C) in your network, you can configure them so that groups B and C can communicate with group A without restriction, but groups B and C cannot communicate between each other.You can also configure specific members of the same group to not accept calls from members of the same group.

The basis for CUGs are interlock codes. Interlock codes are:

- Unique in the whole network. Members belonging to a CUG are assigned a unique interlock code. Members of CUGs will use this interlock code while communicating with other members of the same or different CUGs.
- Passed in CUG interlock code information element (CUG IC IE). The CUG IE also carries information that specifies whether the call can go through if the called party is not a member of the specified CUG.

At the network boundary where the call originates, when a call is received from the user, the DSLAM or switch generates the CUG IE and sends it as part of the SETUP message. In this software release, the CUG IE can only contain the preferential CUG's interlock code. The CUG IE is used at the destination network interface to determine if the call should be forwarded or rejected. The CUG IE is forwarded transparently by the intermediate DSLAMs or switches.

Note End systems do not have any knowledge of interlock codes.

Two types of interlock codes are defined:

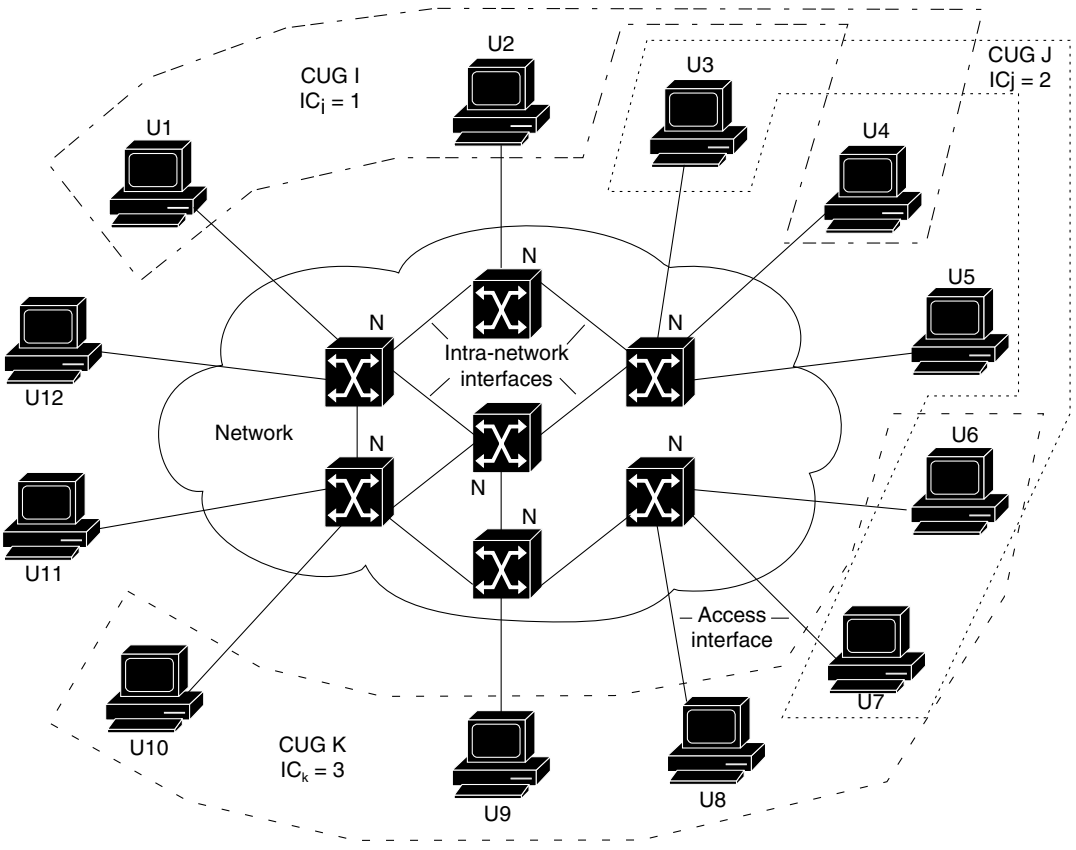
- Global interlock code is 24 bytes long and consists of a globally unique ATM End System Address (AESA) used to identify the network administering the CUG, followed by a 4-byte suffix assigned to this CUG by the network administration.
- International interlock code is 4 bytes long and consists of 4 binary coded decimal (BCD) digits containing a country code and network code, followed by a 2-byte suffix assigned to this CUG by the network administration.

Note Cisco only supports the 24-byte interlock code.

Figure 15-5 provides examples of CUGs and consists of these components:

- Members of CUG I: U1,U2, U4
- Members of CUG J: U3, U6, U7
- Members of CUG K: U6, U7, U8, U9, U10
- U11, U12 do not belong to any closed user groups
- Some of the members of CUG J (U6 and U7) also belong to CUG K

Figure 15-5 Closed User Groups



Two CUG calls shown in Figure 15-5 are:

- A call from U1 to U10 is an inter-CUG call since both users belong to different groups with different CUG interlock codes. The call is rejected at the switch connected to U10 if either the interface to U10 is not configured to accept calls *from* other groups, or the interface from the originating switch to U1 is not configured to allow origination of calls *to* other groups.
- A call from U1 to U2 is an intra-CUG call, since both users belong to the same group with the same CUG interlock code. The call is accepted at the switch connected to U2, unless the configuration of CUG I on the interface to U2 specifies that calls *from* the same group should not be accepted.

Configuring Aliases for CUG Interlock Code

You can define an alias for each CUG interlock code used on the DSLAM. Using an alias can simplify configuration of a CUG on multiple interfaces. When you use an alias, you no longer need to specify the 48-hexadecimal-digit CUG interlock code on each interface attached to a CUG member.

To configure an alias for a CUG interlock code, use this command in global configuration mode:

Command	Task
atm signalling cug alias *alias_name* **interlock-code** *interlock_code*	Configure the alias for the CUG interlock code.

Example

This example shows how to configure the alias TEST for the CUG interlock code 47009181000000000603E5A790100603E5A790100.12345678:

```
DSLAM(config)# atm signalling cug alias TEST interlock-code
47009181000000000603E5A790100603E5A790100.12345678
```

Configuring CUG on an Interface

This section describes how to configure CUG on interfaces.

To perform CUG configuration:

Step 1 Identify the *access interfaces.* Transmission and reception of CUG interlock codes is not allowed over access interfaces. Configuring all interfaces leading outside of the network as access interfaces ensures that all CUG interlock codes are generated and used only within this network. You implement CUG procedures only if you configure the interface as an access interface.

Step 2 Configure each access interface to permit or deny calls either *from* users attached to this interface or *to* unknown users that are not members of this interface's CUGs. In International Telecommunications Union Telecommunications Standardization Sector (ITU-T) terminology, this is called *outgoing access.* Similarly, each access interface can be configured to permit or deny calls either *to* the users attached to this interface or *from* unknown users that are not members of this interface's CUGs. In ITU-T terminology, this is called *incoming access.*

Note Interfaces to other networks should be configured as CUG access interfaces, even if no CUGs are configured on the interface. In this case, if you want the DSLAM to exchange SVCs with the neighbor network, calls *to* and *from* unknown users should be permitted on the interface.

Step 3 Configure each access interface to have one or more CUGs associated with it, but only one CUG can be selected as the *preferential* CUG. In this software release, calls received *from* users attached to this interface can only be associated with the preferential CUG. Calls destined *to* users attached to this interface can be accepted based on membership in any of the CUGs configured for the interface.

Note You can configure CUG service without any preferential CUG. If a preferential CUG is not configured on the interface, and calls *from* users attached to this interface *to* unknown users are permitted, the calls will proceed as non-CUG calls, without generating any CUG IEs.

For each CUG configured on the interface, you can specify that calls *to* or *from* other members of the same CUG be denied. In ITU-T terminology, this is called *outgoing-calls-barred* (OCB) and *incoming-calls-barred* (ICB), respectively.

Table 15-2 describes the relationship between the ITU-T CUG terminology and Cisco CUG terminology.

Table 15-2 Cisco CUG and ITU-T CUG Terminology Conversion

ITU-T CUG Terminology	Cisco CUG Terminology
preferential CUG	preferential
incoming access allowed	permit-unknown-cugs to-user
outgoing access allowed	permit-unknown-cugs from-user
incoming calls barred (ICB)	deny-same-cug to-user
outgoing calls barred (OCB)	deny-same-cug from-user

To configure an access interface and the CUG in which the interface is a member, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.
2.	**atm signalling cug access [permit-unknown-cugs** {**to-user** | **from-user permanent** | **both-direction permanent**}]	Configure the interface as a CUG access interface.
3.	**atm signalling cug assign** {**alias** *alias_name* | **interlock-code** *interlock_code*} [**deny-same-cug** {**to-user** | **from-user**}] [**preferential**]	Configure the CUG where this interface is a member.

Example

This example shows how to configure an interface as a CUG access interface and assign a preferential CUG and displays the result:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# atm signalling cug access permit-unknown-cugs both-direction permanent
DSLAM(config-if)# atm signalling cug assign interlock-code 4700918100000000603E5A790100603E5A790100.12345678
preferential
```

To display the global CUG configuration, use these EXEC commands:

Command	Task
show atm signalling cug [**interface atm** *slot/port*] [**access** | **alias** *alias-name* | **interlock-code** *interlock_code*]	Display the CUG interface configuration status.
show running-config	Display the CUG global configuration status.

Examples

Example	Task
DSLAM# **show atm signalling cug** Interface: ATM0/1 Cug Alias Name: Cug Interlock Code: 4700918100000000603E5A790100603E5A790100.12345678 Non preferential Cug Permit Network to User Calls Permit User to Network Calls	Display the global CUG configuration using the **show atm signalling cug** EXEC command.

Example	Task
DSLAM# **show atm signalling cug access** Closed User Group Access Interface Parameters: Interface: ATM0/1 Network To User (incoming) access: Permit calls from unknown CUGs to User User To Network (outgoing) access: Permit permanent calls to unknown groups	Display the global CUG access configuration using the **show atm signalling cug access** command.
DSLAM# **show running-config** Building configuration... Current configuration: ! version XX.X no service pad service udp-small-servers service tcp-small-servers ! hostname DSLAM ! ! atm signalling cug alias TEST interlock-code 47.009181000000061705BDA01.0061705BDA01.00.12345678 ! atm address 47.0091.8100.0000.0061.705b.da01.0061.705b.da01.00 <information deleted> ! interface ATM0/0 atm signalling cug access permit-unknown-cugs both-direction permanent ! <information deleted>	Display the CUG global configuration using the **show running-config** command.

To display the ATM signaling statistics, use the EXEC command:

Command	Task
show atm signalling statistics	Display the ATM signaling statistics.

Example

This example displays the ATM signaling statistics:

```
DSLAM# show atm signalling statistics
Global Statistics:
Calls Throttled: 0
Max Crankback: 3
Max Connections Pending: 255
Max Connections Pending Hi Water Mark: 1
ATM 0/0:0   UP Time 01:06:20  # of int resets: 0
----------------------------------------------------------------
Terminating connections: 0       Soft VCs: 0
Active Transit PTP SVC: 0        Active Transit MTP SVC: 0
Port requests:  0                Source route requests: 0
```

```
Conn-Pending: 0                  Conn-Pending High Water Mark: 1
Calls Throttled: 0               Max-Conn-Pending:  40
          Messages:   Incoming  Outgoing
          ---------   --------  --------
PTP Setup Messages:         0         0
MTP Setup Messages:         0         0
  Release Messages:         0         0
  Restart Messages:         0         0
           Message:   Received Transmitted Tx-Reject Rx-Reject
Add Party Messages:          0           0         0         0
     Failure Cause:   Routing     CAC   Access-list    Addr-Reg  Misc-Failure
    Location Local:         0       0             0           0         12334
   Location Remote:         0       0             0           0             0
ATM 0/2:0   UP Time 3d21h  # of int resets: 0
------------------------------------------------------------------
Terminating connections: 0       Soft VCs: 0
Active Transit PTP SVC: 0        Active Transit MTP SVC: 0
Port requests:  0                Source route requests: 0
Conn-Pending: 0                  Conn-Pending High Water Mark: 0
Calls Throttled: 0               Max-Conn-Pending:  40

<information deleted>
```

Disabling Signaling on an Interface

If you disable signaling on a PNNI interface, PNNI routing is also disabled and ILMI is automatically restarted each time signaling is enabled or disabled.

To disable signaling on an interface, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	**interface atm** *slot/port*	Select the interface to be configured.
2.	**no atm signalling enable**	Disable signaling on the interface.

Example

This example shows how to disable signaling on ATM interface 0/1:

```
DSLAM(config)# interface atm 0/1
DSLAM(config-if)# no atm signalling enable
DSLAM(config-if)#
%ATM-5-ATMSOFTSTART: Restarting ATM signalling and ILMI on ATM0/1.
```

CHAPTER 16

Configuring the Trunk and Subtended Interfaces

This chapter describes the steps required to configure the trunk and subtended interfaces on the Cisco DSLAM NI-2 card and includes these sections:

NI-2 Card and DSLAM Compatibility

The following shows the NI-2 card and DSLAM chassis compatibility with regard to both trunk and subtending connections.

NI-2 Card	Cisco 6015	Cisco 6100 / Cisco 6130	Cisco 6160	Cisco 6260
DS3+T1/E1 IMA[1] • DS3 trunk • T1/E1 trunk and subtending • T1/E1 IMA trunk and subtending	Yes	No	Yes[2]	No
DS3/2DS3 • DS3 trunk • two DS3 subtending ports	No	Yes	Yes	Yes[3]
OC-3c/OC-3c single-mode fiber (SMF) • OC-3c trunk • one OC-3c subending port	No	Yes	Yes	Yes[4]
OC-3c/OC-3c multimode fiber (MMF) • OC-3c trunk • one OC-3c subending port	No	Yes	Yes	Yes[4]

NI-2 Card	Cisco 6015	Cisco 6100 / Cisco 6130	Cisco 6160	Cisco 6260
OC-3c/2DS3 single-mode fiber (SMF) • OC-3c trunk • two DS3 subtending ports	No	No	Yes	No
OC-3c/2DS3 multimode fiber (MMF) • OC-3c trunk • two DS3subending ports	No	No	Yes	No

1. inverse multiplexing over ATM.
2. Use only with the DS3/2DS3+8xT1 system I/O card.
3. When the E3 1/0 module is installed, the system assumes E3 funtionality.
4. When the OC-3c I/O module is installed, the system assumes OC-3c functionality.

NI-2 Subtending Support

NI-2 cards offer the same level of service and traffic fairness in subtending Cisco 6015, Cisco 6100, Cisco 6130, Cisco 6160, and Cisco 6260 nodes. The level of service remains the same for both NI-1 and NI-2 based subtended nodes. (That is, you can mix NI-1 and NI-2 cards in the same subtending network for the Cisco 6100 and Cisco 6130 chassis.)

The following guidelines apply to subtending on an NI-2 supported DSLAM:

- For the Cisco 6100 and Cisco 6130, the NI-2 accepts the same virtual path (VP) and virtual circuit (VC) constraints that exist on the NI-1.
- The NI-2 allows subtending for up to 1664 ports per system.
  - The Cisco 6015 has one subtend host chassis and up to six subtended node chassis.
  - The other chassis have one subtend host chassis and up to twelve subtended node chassis.
- The NI-2 supports tree and daisy chain subtending.

Configuring 155 Mbps OC-3 SM and MM Interfaces

You can configure the NI-2 ports as redundant links using the switch routing protocols. The NI-2 card supports system controller-type connectors.

Each port can be configured to support these clocking options:

- Self-timing based on a Stratum three-level clock
- Loop timing from the received data stream—Ideal for public network connections
- Timing synchronized to a selected master clock port—Required to distribute a single clock across a network

Traffic pacing allows you to set the aggregate output traffic rate on any port to a rate below the line rate. This feature is useful when communicating with a slow receiver or when connected to public networks with peak-rate tariffs.

The plug-and-play mechanisms of the DSLAM allow the interface to launch automatically. You can save all configuration information between hot swaps and reboots, while interface types are automatically discovered by the DSLAM, eliminating the need for mandatory manual configuration.

Default 155 Mbps ATM Interface Configuration Without Autoconfiguration

If Interim Local Management Interface (ILMI) has been disabled or if the connecting end node does not support ILMI, these defaults are assigned to all 155 Mbps (OC-3c) interfaces:

- ATM interface type = user network interface (UNI)
- UNI Version = 3.1
- Maximum virtual path identifier (VPI) bits = 8
- Maximum virtual channel identifier (VCI) bits = 14
- ATM interface side = network
- ATM UNI type = private
- Framing = sts-3c
- Clock source = network-derived
- Synchronous Transfer Signal (STS) stream scrambling = enabled
- Cell payload scrambling = enabled

The default subtend ID for each NI-2 DSLAM is 0 (zero).

Manual 155 Mbps Interface Configuration

To manually change any of the default configuration values, perform these tasks, beginning in global configuration mode.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**subtend-id** *0-12*	Assign to this node a subtend ID that is unique in the subtend tree. The node attached to the trunk must have subtend ID 0.
3.	DSLAM(config)#**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.
4.	DSLAM(config-if)#**atm uni** [**side** {**network** | **user**} **type** {**private** | **public**} **version** {**3.0** | **3.1** | **4.0**}]	Modify the ATM interface side, type, or version.
5.	DSLAM(config-if)#**atm maxvpi-bits** *0-8*	Modify the maximum VPI bits configuration.
6.	DSLAM(config-if)#**atm maxvci-bits** *0-14*	Modify the maximum VCI bits configuration.
7.	DSLAM(config-if)#**sonet** {**stm-1** | **sts-3c**}	Modify the framing mode.
8.	DSLAM(config-if)#**clock source** {**free-running** | **loop-timed** | **network-derived**}	Modify the clock source.

Step	Command	Task
9.	DSLAM(config-if)#**scrambling** {**cell-payload** | **sts-stream**}	Modify the scrambling mode.
10.	DSLAM(config-if)#**exit**	Return to global configuration mode.
11.	DSLAM(config)#**subtend-id** *0-12*	Assign to this interface a subtend ID that is unique in the subtend tree. (This subtend ID identifies the subtended node attached to the interface, in the case where the attached node does not support the subtend ID feature.)

Note Note that Steps 1 and 9 are alternatives; do not perform both steps.

Examples

This example shows how to change the default ATM interface type to **private** using the **atm uni type private** command.

```
DSLAM#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)#interface atm 0/0
DSLAM(config-if)#atm uni type private
```

This example shows how to change the clock source using the **clock source network-derived** command.

```
DSLAM#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)#interface atm 0/0
DSLAM(config-if)#clock source network-derived
```

Note Refer to the "Interface Configuration Troubleshooting" section on page 16-7, to confirm your interface configuration.

Configuring DS3 and E3 Interfaces

Use the 45-Mbps DS3 to:

- Set up wide-area connections
- Link multiple campuses
- Connect to public networks

The ports can be set up as redundant links for use by sophisticated switch routing protocols.

You can configure each port to support these clocking options:

- Self-timing based on a Stratum 4 level clock
- Loop timing from the received data stream—Ideal for public network connections
- Timing synchronized to a selected master clock port—Required to distribute a single clock across a network

Note Network clocking configuration options are applicable only to DS3 quad interfaces.

Traffic pacing allows you to set the aggregate output traffic rate on any port to a rate below the line rate. This feature is useful when communicating with a slow receiver or when connected to public networks with peak-rate tariffs.

The plug-and-play mechanisms of the DSLAM allow the interface to launch automatically. You can save all configuration information between hot swaps and reboots, while interface types are automatically discovered by the DSLAM, eliminating the need for mandatory manual configuration.

Default DS3 ATM Interface Configuration Without Autoconfiguration

If ILMI has been disabled or if the connecting end node does not support ILMI, these defaults are assigned to all DS3 interfaces:

- ATM interface type = UNI
- UNI Version = 3.0
- Maximum VPI bits = 8
- Maximum VCI bits = 14
- ATM interface side = network
- ATM UNI type = private

These defaults are assigned to all DS3 interfaces:

- Framing = cbit-adm
- Cell payload scrambling = disabled
- Clock source = network-derived
- Electrical line build out (LBO) = short
- Auto-far-end remote failure(ferf) on loss of signal (LOS) = on
- Auto-ferf on out of frame (OOF) = on
- Auto-ferf on red = on
- Auto-ferf on loss of cell delineation (LCD) = on
- Auto-ferf on alarm indication signaling (AIS) = on

These defaults are assigned to all E3 interfaces:

- Framing = G.832 adm
- Cell payload scrambling = on
- Clock source = network-derived
- Auto-ferf on LOS = on
- Auto-ferf on OOF = on
- Auto-ferf on LCD = on (applicable to nonplcp mode only)
- Auto-ferf on AIS = on

The default subtend ID for each NI-2 DSLAM is 0 (zero).

Manual DS3 and E3 Interface Configuration

To manually change any of the DS3 or E3 default configuration values, perform these tasks, beginning in global configuration mode.

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
1.	DSLAM(config)#**subtend-id** *0-12*	Assign to this node a subtend ID that is unique in the subtend tree. The node attached to the trunk must have subtend ID 0.
2.	DSLAM(config)#**interface atm** *slot/port*	Specify an ATM interface and enter interface configuration mode.
3.	DSLAM(config-if)#**atm uni** [**side** {**network** | **user**} **type** {**private** | **public**} **version** {**3.0** | **3.1** | **4.0**}]	Modify the ATM interface side, type, or version.
4.	DSLAM(config-if)#**atm maxvpi-bits** *0-8*	Modify the maximum VPI bits configuration.
5.	DSLAM(config-if)#**atm maxvci-bits** *0-14*	Modify the maximum VCI bits configuration.
6.	DSLAM(config-if)#**framing** {**cbitadm** | **cbitplcp** | **m23adm** | **m23plcp**}	Modify the framing mode.
7.	DSLAM(config-if)#**scrambling** {**cell-payload** | **sts-stream**}	Modify the scrambling mode.
8.	DSLAM(config-if)#**clock source** {**free-running** | **loop-timed** | **network-derived**}	Modify the clock source.
9.	DSLAM(config-if)#**network-clock-select** {*1-4_priority*} **atm** *slot/port*	Configure the network-derived clock.
10.	DSLAM(config-if)#**lbo** {**long** | **short**}	Modify the line build-out.
11.	DSLAM(config-if)#**auto-ferf** {**ais** | **lcd** | **los** | **oof** | **red**}	Modify the auto-ferf configuration.
12.	DSLAM(config-if)#**exit**	Return to global configuration mode.
13.	DSLAM(config)#**subtend-id** *0-12*	Assign to this interface a subtend ID that is unique in the subtend tree. (This subtend ID identifies the subtended node attached to the interface, in the case where the attached node does not support the subtend ID feature.)

Note Note that Steps 1 and 9 are alternatives; do not perform both steps.

Examples

This example shows how to change the default ATM interface type to **private** using the **atm uni type private** command.

```
DSLAM#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)#interface atm 0/0
DSLAM(config-if)#atm uni type private
```

This example shows how to change the clock source using the **clock source network-derived** command.

```
DSLAM#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)#interface atm 0/0
DSLAM(config-if)#clock source network-derived
```

Note Refer to the "Interface Configuration Troubleshooting" section on page 16-7, to confirm your interface configuration.

Interface Configuration Troubleshooting

The following are privileged EXEC mode commands that you can use to confirm that the hardware, software, and interfaces for the DSLAM are configured as intended.

Command	Description
DSLAM#**show version**	Confirm the correct version and type of software is installed.
DSLAM#**show hardware**	Confirm the type of hardware installed in the system.
DSLAM#**show interface ethernet [slot/port]**	Confirm the type of hardware installed in the system.
DSLAM#**show atm addresses**	Confirm the ATM address is configured correctly.
DSLAM#**ping atm interface atm [slot/port] [vpi] ip-address xxx.xxx.xxx.xxx**	Test for connectivity between the DSLAM and a host.
DSLAM#**show** {**atm** | **ces**} **interface**	Confirm the ATM interfaces are configured correctly.
DSLAM#**show atm status**	Confirm the status of the ATM interfaces.
DSLAM#**show atm vc**	Confirm the status of ATM virtual interfaces.
DSLAM#**show running-config**	Confirm the configuration being used is configured correctly.
DSLAM#**show startup-config**	Confirm the configuration saved in NVRAM is configured correctly.
DSLAM#**show controller** {**atm** | **ethernet**}	Confirm interface controller memory addressing.

You can also view an ATM layer fault state and loss of cell delineation using the CLI and MIB. The default alarm level for this fault state is major.

The following are privileged EXEC mode commands you can use to initiate line loopbacks.

Command	Description
DSLAM#**loopback diagnostic**	Diagnostic loopback. The outgoing cells are looped back toward the switch. This command is available on all ports.
DSLAM#**loopback line**	Line loopback. The incoming line is looped back toward the coax. This command is available only on trunk and subtending ports.
DSLAM#**loopback payload**	Payload loopback: The incoming payload is looped back toward the coax. This command is available only on DS3 trunk and subtending ports.

CHAPTER 17

Loading System Software Images and Configuration Files

This chapter describes how to load and maintain system software images and configuration files for Cisco DSLAMs with NI-2. The instructions in this chapter assume that your DSLAM contains a minimal configuration that allows you to interact with the system software.

The tasks in the first four sections are typical tasks for all DSLAMs:

Configuring a Static IP Route

If you are managing the DSLAM through an Ethernet interface or ATM subinterface on the ATM switch processor (ASP), and your management station or Trivial File Transfer Protocol (TFTP) server is on a different subnet than the DSLAM, you must first configure a static IP route.

Caution If you do not configure a static IP route before you install the new image, this results in a loss of remote administrative access to the DSLAM. If this happens, you can regain access from a direct console connection to the DSLAM, although this requires physical access to the console port.

To configure a static IP route, follow these steps:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**ip route** *prefix*[1] *mask*[2] **[ethernet | atm] 0/0[.***subinterface***]**	Configure a static IP route on the Ethernet interface or ATM subinterface of the ASP.
3.	DSLAM(config)#**end**	Return to privileged EXEC mode.
4.	DSLAM#**copy running-config startup-config**	Save the configuration to NVRAM.

1. The IP route prefix of the remote network in which the management station or TFTP server resides.
2. The subnet mask of the remote network in which the management station or TFTP server resides.

Retrieving System Software Images and Configuration Files

If you have a minimal configuration that allows you to interact with the system software, you can retrieve other system images and configuration files from a network server and modify them for use in your particular routing environment. To retrieve system images and configuration files for modification, perform the tasks described in this section.

Copying System Software Images from a Network Server to the DSLAM

You can copy system images from a TFTP, Remote Copy Protocol (rcp), or Maintenance Operation Protocol (MOP) server to the DSLAM's Flash memory. The DSLAM uses embedded Flash memory.

Using Flash Memory

In Flash memory, if free space is:

- Available in Flash memory, you can erase the existing Flash memory before writing onto it.
- Not available, or if the Flash memory has never been written to, the format routine is required before new files can be copied.

The system informs you of these conditions and prompts you for a response. If you accept the erasure, the system prompts you again to confirm before erasing.

Note The Flash memory is erased at the factory before shipment.

If you attempt to copy a file that already exists into Flash memory, a prompt informs you that a file with the same name already exists. The older file is deleted when you copy the new file into Flash. The first copy of the file still resides within Flash memory, but it is made unusable in favor of the newest version, and is listed with the "deleted" tag when you use the **show flash** command. If you terminate the copy process, the newer file is marked "deleted" because the entire file was not copied. In this case, the original file in Flash memory is valid and available to the system.

Note You can copy normal system images or system images compressed with the UNIX **compress** command to Flash memory.

Copying from a TFTP Server to Flash Memory

To copy a system image from a TFTP server to Flash memory, follow these steps:

Step	Command	Task
1.	DSLAM>**enable** Password:	Go to privileged EXEC mode.
2.	DSLAM#**cd bootflash**	Change directory to bootflash, the embedded Flash directory.
3.	See the instructions in the section "Copying System Images from Flash Memory to a Network Server" section on page 17-25.	Make a backup copy of the current system software image.
4.	DSLAM#**copy tftp flash** DSLAM#**copy tftp** *file_id*	Copy a system image to Flash memory.
5.	*ip-address* or *name*	If prompted, enter the IP address or domain name of the server.
6.	*filename*	If prompted, enter the filename of the server system image. filenames are case-sensitive.
7.	*device*	If prompted, enter the Flash memory device that is to receive the copy of the system image.

Note Be sure there is ample space available before copying a file to Flash memory. Use the **dir** command and compare the size of the file you want to copy to the amount of available Flash memory shown. If the space available is less than the space required by the file you want to copy, the copy process continues, but the entire file is not copied into Flash memory. The failure message "buffer overflow - *xxxx/xxxx*" appears, where *xxxx/xxxx* is the number of bytes read in relation to the number of bytes available.

When you issue the **copy tftp flash** command, the system prompts you for the IP address or domain name of the TFTP server. This server can be another switch or DSLAM serving ROM or Flash system software images. The system prompts you for the filename of the software image to copy.

When you issue the copy tftp flash and copy tftp file_id commands, if there is free space available in Flash memory, you are given the option of erasing the existing Flash memory before writing onto it. If no free Flash memory space is available, or if the Flash memory has never been written to, the erase routine is required before new files can be copied. The system informs you of these conditions and prompts you for a response.

The *file_id* argument of the **copy tftp** *file_id* command specifies a device and filename as the destination of the copy operation. You can omit the device and enter only **copy tftp** *filename*. If you omit the device, the system uses the current device specified by the **cd** command. You must choose **bootflash:** as the Flash memory device.

Note Use the **pwd** command to display the current device.

Examples

This example shows how to copy a system image named "6260-wi-m_1.1(1)" into the current Flash configuration:

```
DSLAM#copy tftp flash
Enter source file name: 6260-wi-m_1.1(1)
Enter destination file name [6260-wi-m_1.1(1)]:
7602048 bytes available on device bootflash, proceed? [confirm] y
Address or name of remote host [dirt.cisco.com]?
Accessing file "6260-wi-m_1.1(1)" on dirt.cisco.com ...FOUND
Loading 6260-wi-m_1.1(1) from 171.69.1.129 (via Ethernet0/0): !!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 2247751/4495360 bytes]
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

The exclamation points indicate that the process is working. A series of Cs indicates that a checksum verification of the image is occurring after the image is written to Flash memory.

The **dir** command confirms that the file transfer was successful, as in this example:

```
DSLAM#dir
-#- -length- -----date/time------ name
1   2247751  May 03 2000 14:32:10 6260-wi-m_1.1(1)

5354296 bytes available (2247880 bytes used)
```

This example shows how to copy the "dslam-confg" file from a TFTP server to embedded Flash memory. The copied file has the name "backup-confg".

```
DSLAM#copy tftp:dslam-confg bootflash:backup-confg
1244732 bytes available on device slot0, proceed? [confirm] y
Address or name of remote host [dirt.cisco.com]?
Accessing file "dslam-confg" on dirt.cisco.com ...FOUND
Loading dslam-confg from 171.69.1.129 (via Ethernet0/0): !!
[OK - 5204/10240 bytes]
```

Copying from an rcp Server to Flash Memory

You can copy a system image from an rcp network server to Flash memory.

For the **rcp** command to execute successfully, you must define an account on the network server for the remote username. You can override the default remote username sent on the rcp copy request by configuring the remote username.

For example, if the system image resides in the home directory of a user on the server, you can specify that user's name as the remote username. The rcp protocol implementation copies the system image from the remote server to the directory of the remote username if the remote server has a directory structure, as do UNIX systems.

To copy a system image from an rcp server to Flash memory, follow these steps:

Step	Command	Tasks
1.	See the instructions in the section "Copying System Images from Flash Memory to a Network Server" section on page 17-25.	Make a backup copy of the current system software image.
2.	DSLAM#**configure terminal**	Enter global configuration mode from the terminal. This step is required only if you override the default remote username (see Step 3).
3.	DSLAM(config)#**ip rcmd remote-username** *username*	Specify the remote username.
4.	DSLAM(config)#**end**	Exit global configuration mode.
5.	DSLAM#**copy rcp flash** DSLAM#**copy rcp** *file_id*	Copy the system image from an rcp server to Flash memory.
6.	*ip-address* or *name*	If prompted, enter the IP address or domain name of the network server.
7.	*filename*	If prompted, enter the filename of the server system image to be copied.

The **copy** command automatically displays the Flash memory directory, including the amount of free space. If the file being downloaded to Flash memory is an uncompressed system image, the **copy** command automatically determines the size of the file being downloaded and validates it with the space available in Flash memory.

When you issue the **copy rcp flash** or **copy rcp** *file_id* command, the system prompts you for the IP address or domain name of the server. This server can be another switch or DSLAM serving Flash system software images. The system then prompts you for the filename of the software image to copy. With the **copy rcp flash** command, the system also prompts you to name the system image file that resides in Flash memory after the copy is complete. You can use the filename of the source file, or you can choose another name.

Examples

This example shows how to copy a system image named "mysysim1" from the "netadmin1" directory on the remote server named "SERVER1.CISCO.COM" with an IP address of 171.69.1.129 to the DSLAM's Flash memory. To ensure that enough Flash memory is available to accommodate the system image to be copied, the DSLAM software allows you to erase the contents of Flash memory first.

```
DSLAM#configure terminal
DSLAM(config)#ip rcmd remote-username netadmin1
DSLAM(config)#end
DSLAM#copy rcp flash
Enter source file name: 6260-wi-m_1.1(1)
Enter destination file name [6260-wi-m_1.1(1)]:
3498136 bytes available on device slot0, proceed? [confirm] y
Address or name of remote host [server1.cisco.com]?

Connected to 171.69.1.129
Loading 2247751 byte file 6260-wi-m_1.1(1):
Connected to 171.69.1.129
Loading 2247751 byte file 6260-wi-m_1.1(1): !!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!! [OK]
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

The exclamation points indicate that the process is working.

Note If you enter **n** after the “proceed?” prompt, the copy process stops. If you enter **y** and confirm the copy, copying begins. Make sure there is ample Flash memory available before entering **y** at the proceed prompt.

This example uses the **copy rcp** *file_id* command to copy the “dslam-image” file from a network server using rcp to the embedded Flash memory:

```
DSLAM#configure terminal
DSLAM(config)#ip rcmd remote-username netadmin1
DSLAM(config)#end
DSLAM#copy rcp bootflash:dslam-image
```

Verifying the Image in Flash Memory

Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the system software image. When you issue the **copy tftp flash**, **copy rcp flash**, or **copy rcp bootflash** commands, the checksum of the image in Flash memory appears at the bottom of the screen. The README file was copied to the network server automatically when you installed the system software image on the server.

Caution If the checksum value does not match the value in the README file, do not reboot the DSLAM. Instead, issue the copy request and compare the checksums again. If the checksum is repeatedly incorrect, copy the original system software image back into Flash memory *before* you reboot the DSLAM from Flash memory. If you have a corrupted image in Flash memory and try to boot from Flash, the DSLAM starts the system image contained in ROM (assuming that booting from a network server is not configured). If ROM does not contain a fully functional system image, the DSLAM does not function, and you must reconfigure it using a direct console port connection.

Copying Configuration Files from a Network Server to the DSLAM

You can copy configuration files from:

- A TFTP server or an rcp server to the DSLAM. You might use this process to:
    - Restore a configuration file to the DSLAM if you have backed up the file to a server. If you replace a DSLAM and want to use the configuration file that you created for the original DSLAM, you can restore that file instead of recreating it.
    - Copy to the DSLAM a different configuration that is stored on a network server.
- An rcp or TFTP server to either the running configuration or the startup configuration. When you copy a configuration file to:
    - The running configuration, you copy to and run the file from RAM.

– The startup configuration, you copy it to nonvolatile random-access memory (NVRAM) or to the location specified by the CONFIG_FILE environment variable.

Copying from a TFTP Server to the DSLAM

To copy a configuration file from a TFTP server to the DSLAM, complete these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM>**enable** Password:	Go to privileged EXEC mode.
2.	DSLAM#**copy tftp running-config** or DSLAM#**copy tftp startup-config**	Copy a configuration file from a TFTP server to the DSLAM's running or startup configuration.
3.	*ip-address* or *name*	If prompted, enter the IP address or domain name of the server.
4.	*filename*	If prompted, enter the filename of the server system image.

Copying from an rcp Server to the DSLAM

The rcp protocol requires that a client send the remote username on each rcp request to a network server. When you issue a request to copy a configuration file from an rcp network server, the DSLAM sends a default remote username unless you override the default by configuring a remote username. By default, the DSLAM software sends the remote username associated with the current teletype (TTY) process, if that name is valid. If the TTY username is invalid, the DSLAM software uses the DSLAM host name as both the remote and local user names. You can also specify the path of an existing directory with the remote username.

For the rcp copy request to execute successfully:

Step 1 Define an account on the network server for the remote username.

Step 2 If you copy the configuration file from a personal computer used as a file server, make sure that the remote host computer supports the remote shell protocol.

To copy a configuration file from an rcp server to the running configuration or the startup configuration, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(configure)#**ip rcmd remote-username** *username*	Specify the remote username. This step is optional, but recommended.
3.	DSLAM(configure)#**end**	Exit configuration mode.

Step	Command	Task
4.	DSLAM#**copy rcp running-config** or DSLAM#**copy rcp startup-config**	Copy a configuration file from an rcp server to the DSLAM's running or startup configuration.
5.	*ip-address*	If prompted, enter the IP address of the server.
6.	*filename*	If prompted, enter the name of the configuration file.

The **copy rcp startup-config** command copies the configuration file from the network server to the configuration file pointed to by the CONFIG_FILE environment variable. If you want to write the configuration file from the server to NVRAM on the DSLAM, be sure to set the CONFIG_FILE environment variable to NVRAM. Refer to the "Downloading the CONFIG_FILE Environment Variable Configuration" section on page 17-23 in this chapter for instructions on setting the CONFIG_FILE environment variable.

Examples

Using the remote username "netadmin1", this example shows copying a host configuration file "host1-confg" from the "netadmin1" directory on the remote server to the DSLAM's startup configuration:

```
DSLAM#configure terminal
DSLAM(config)#ip rcmd remote-username netadmin1
DSLAM(config)#end
DSLAM#copy rcp running-config
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 131.108.101.101
Name of configuration file [dslam-confg]? host1-confg
Configure using host1-confg from 131.108.101.101? [confirm]
Connected to 131.108.101.101
Loading 1112 byte file host1-confg:![OK]
DSLAM#
%SYS-5-CONFIG: Configured from host1-config by rcp from 131.108.101.101
```

Using the remote username "netadmin1", this example shows copying a host configuration file "host2-confg" from the "netadmin1" directory on the remote server to the DSLAM's startup configuration:

```
DSLAM#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DSLAM(config)#ip rcmd remote-username netadmin1
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM#copy host2-confg rcp
Remote host []? dirt
Name of configuration file to write [dslam-confg]?
Write file dslam-confg on host 171.69.1.129? [confirm]
Writing dslam-confg !! [OK]
DSLAM#copy rcp startup-config
Address of remote host [255.255.255.255]? 171.69.1.129
Name of configuration file [dslam-confg]?
Configure using dslam-confg from 171.69.1.129? [confirm]

Connected to 171.69.1.129
Loading 5393 byte file dslam-confg: !! [OK]
```

```
Warning: distilled config is not generated
[OK]
DSLAM#
%SYS-5-CONFIG_NV: Non-volatile store configured from dslam-confg by console rcp
 from 171.69.1.129
```

Changing the Buffer Size for Loading Configuration Files

The buffer that holds the configuration commands is generally the size of NVRAM. Complex configurations may require a larger configuration file buffer size. To change the buffer size, use this command in global configuration mode:

Command	Task
DSLAM(config)#**boot buffersize** *bytes*	Change the buffer size to use for booting a host or network configuration file from a network server.

Example

In this example, the buffer size is set to 50000 bytes, and the running configuration is saved to the startup-configuration:

```
DSLAM(config)#boot buffersize 50000
DSLAM(config)#end
DSLAM#copy running-config startup-config
Destination filename [startup-config]? y
Building configuration...
[OK]
```

Displaying System Image and Configuration Information

To display information about system software, system image files, and configuration files, use these privileged EXEC commands:

Command	Task
DSLAM#**show version**	List the system software release version, configuration register setting, and so on.
DSLAM#**show boot**	List the contents of the BOOT environment variable, the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
DSLAM#**show startup-config**	List the startup configuration information. The CONFIG_FILE environment variable points to the startup configuration.
DSLAM#**show file** *device*:*filename*	List the configuration information stored in a specified file.

Command	Task
DSLAM#**show running-config**	List the configuration information in running memory.
DSLAM#**show flash**	List information about Flash memory, including system image filenames and amounts of memory used and remaining.

You can also use the **o** command in ROM monitor mode to list the configuration register settings.

Performing General Startup Tasks

If you modify your switching environment, you must perform some general startup tasks. For example, to modify a configuration file, you enter configuration mode. You also modify the configuration register boot field to tell the DSLAM if and how to load a system image upon startup. Also, instead of using the default system image and configuration file to start up, you can specify a particular system image and configuration file for the DSLAM to use to start up.

General startup tasks include:

- Enter Configuration Mode and Select a Configuration Source
- Modify the Configuration Register Boot Field
- Specify the Startup System Image
- Specify the Startup Configuration File

Entering Configuration Mode and Select a Configuration Source

When you enter configuration mode using the **configure** privileged EXEC command, you must specify the source of the configuration as **terminal**, **memory**, **network**, or **overwrite-network**. Each of these methods is described in these subsections.

The DSLAM accepts one configuration command per line. You can:

- Enter as many configuration commands as you want.
- Add comments to a configuration file by placing an exclamation point (!) at the beginning of each comment line. Comments, as well as default settings, are *not* stored in NVRAM or in the active copy of the configuration file and therefore do not appear when you list the active configuration with the **show running-config** EXEC command or the startup configuration with the **show startup-config** EXEC command (when the startup configuration is stored in NVRAM). However, you can list the comments in configuration files stored on a TFTP, rcp, or MOP server.

Configuring the DSLAM from the Terminal

When you configure the DSLAM from the terminal, you do so interactively: the DSLAM executes the commands as you enter them at the system prompts. To configure the DSLAM from the terminal, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	See the appropriate chapter for specific configuration commands.	Enter the necessary configuration commands.
3.	DSLAM(config)#**end**	Quit configuration mode.
4.	DSLAM#**copy running-config startup-config**	Save the configuration file to your startup configuration. This step saves the configuration to the location specified by the CONFIG_FILE environment variable.

Example

In this example, the DSLAM is configured from the terminal. The **hostname** command changes the DSLAM name to "dslam2". The **end** command quits configuration mode, and the **copy running-config startup-config** command saves the current configuration to the startup configuration. The next time you start up the DSLAM the host name will be "dslam2".

```
DSLAM#configure terminal
DSLAM(config)#hostname dslam2
DSLAM(config)#end
DSLAM#copy running-config startup-config
```

Configuring the DSLAM from Memory

When you configure the DSLAM from memory, the DSLAM executes the commands in NVRAM, or the configuration specified by the CONFIG_FILE environment variable. To configure from memory, use this command in privileged EXEC mode:

Command	Task
DSLAM#**configure memory**	Configure the DSLAM to execute the configuration specified by the CONFIG_FILE environment variable or NVRAM.

For an explanation of the CONFIG_FILE environment variable, see the "Downloading the CONFIG_FILE Environment Variable Configuration" section on page 17-23.

Configuring the DSLAM from the Network

To configure the DSLAM by retrieving a configuration file stored on one of your network servers, perform these tasks, beginning in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**configure network**	Enter configuration mode with the network option.
2	**host** or **network**	At the system prompt, select a network or host configuration file. The network configuration file contains commands that apply to all network servers and terminal servers on the network. The host configuration file contains commands that apply to only one network server.
3	*ip-address*	At the system prompt, enter the optional IP address of the remote host from which you are retrieving the configuration file.
4	*filename*	At the system prompt, enter the name of the configuration file or accept the default name.
5	**y**	Confirm the configuration filename that the system supplies.

Example

In this example, the DSLAM is configured from the file *backup-config* at IP address 171.69.1.129:

```
DSLAM#configure network
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 171.69.1.129
Name of configuration file [dslam-confg]? backup-confg
Configure using backup-confg from 171.69.1.129? [confirm] y
DSLAM#
%SYS-5-CONFIG: Configured from backup-confg by console tftp from 171.69.1.129
```

Copying a Configuration File Directly to the Startup Configuration

You can copy a configuration file directly to your startup configuration without affecting the running configuration. This process loads a configuration file directly into NVRAM or into the location specified by the CONFIG_FILE environment variable without affecting the running configuration.

To copy a configuration file directly to the startup configuration, perform these tasks, beginning in privileged EXEC mode:

Step	Command	Task
1	DSLAM#**configure overwrite-network** *I received the message, "This command has been replaced by the command: 'copy <url> nvram:/startup-config'"*	Enter configuration mode with the network option.
2	**host** or **network**	At the system prompt, select a network or host configuration file. The network configuration file contains commands that apply to all network servers and terminal servers on the network. The host configuration file contains commands that apply to only one network server.
3	*ip-address*	At the system prompt, enter the optional IP address of the remote host from which you are retrieving the configuration file.
4	*filename*	At the system prompt, enter the name of the configuration file or accept the default name.
5	**y**	Confirm the configuration filename that the system supplies.

Modifying the Configuration Register Boot Field

The configuration register boot field determines whether the DSLAM loads an operating system image, and if so, where it obtains this system image. This section describes how the DSLAM uses the configuration register boot field and how to set and modify this field.

Using the Boot Field

The lowest four bits of the 16-bit configuration register (bits 3, 2, 1, and 0) form the boot field. These boot field values determine if the DSLAM loads an operating system and where the DSLAM obtains the system image:

- When the entire boot field equals 0-0-0-0, the DSLAM does not load a system image. Instead, the DSLAM enters ROM monitor or maintenance mode, from which you can enter ROM monitor commands to manually load a system image.
- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the DSLAM loads the system image specified by **boot system** commands in the startup configuration file. When the startup configuration file does not contain **boot system** commands, the DSLAM loads a default system image stored on a network server.

When you load a default system image from a network server, the DSLAM uses the configuration register settings to determine the default system image filename for booting from a network server. The default boot filename starts with the string "cisco", followed by the octal equivalent of the boot field number in the configuration register, followed by a hyphen (-) and the processor type name (for example, "cisco nn-cpu").

Setting the Boot Field

You must correctly set the configuration register boot field to ensure that your DSLAM loads the operating system image correctly. See the Table 17-1 for boot field descriptions.

Table 17-1 Boot Field Descriptions

Configuration Register	Break Enabled/Disabled[1]	Description
0x000	Enabled	Boot manually.
0x001	Enabled	Boot from ROM.
0x002 through 0x00F	Enabled	Boot from the default filename specified "*nn*" in boot system configuration.
0x100	Disabled	Boot manually.
0x101	Disabled	Boot from ROM.
0x102 through 0x10F	Disabled	Boot from the default filename specified "*nn*" in boot system configuration.

1. Enabled allows a hardware break during the first 30 seconds.

To set the boot field, follow this general procedure:

Step 1 Obtain the current configuration register setting, a hexadecimal value.

Step 2 Modify the current configuration register setting to reflect how you want the DSLAM to load a system image. To do so, change the least significant hexadecimal digit to one of these values:

- 0—Loads the system image manually using the **boot** command in ROM monitor mode.
- 1—Loads the system image from boot ROM.
- 2 to F—Loads the system image from **boot system** commands in the startup configuration file or from a default system image stored on a network server.

For example, if the current configuration register setting is 0x101 and you want to load a system image from **boot system** commands in the startup configuration file, change the configuration register setting to 0x102.

Step 3 Reboot the DSLAM to make your changes to the configuration register take effect.

Performing the Boot Field Modification Tasks

Use the hardware configuration register to modify the boot field of a DSLAM.

To modify the configuration register boot field, complete these tasks, beginning in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**show version**	Obtain the current configuration register setting.
2.	DSLAM#**configure terminal**	Enter global configuration mode, selecting the terminal option.
3.	DSLAM(config)#**config-register** *value*	Modify the existing configuration register setting to specify how you want the DSLAM to load a system image.
4.	DSLAM(config)#**end**	Exit configuration mode.
5.	DSLAM#**reload**	Reboot the DSLAM to make your changes take effect.

In ROM monitor mode, use the **o** command to list the value of the configuration register boot field.

Example

In this example, the **show version** command indicates that the current configuration register is set so that the DSLAM does not automatically load an operating system image. Instead, it enters ROM monitor mode and waits for user-entered ROM monitor commands. The new setting instructs the DSLAM to load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
DSLAM#show version
Cisco Internetwork Operating System Software

<information deleted>

8192K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x0

DSLAM#configure terminal
DSLAM(config)#config-register 0x010F
```

Specifying the Startup System Image

You can enter multiple boot commands in the startup configuration file or in the BOOT environment variable to provide backup methods for loading a system image onto the DSLAM. There are two ways to load a system image:

- From Flash memory—Flash memory allows you to copy new system images without changing erasable programmable read-only memory (EPROM) integrated circuits (ICs). Information stored in Flash memory is not vulnerable to network failures that might occur when loading system images from servers.
- From a network server—If Flash memory becomes corrupted, specifying a system image to be loaded from a network server using TFTP, rcp, or MOP provides a backup boot method for the DSLAM. You can specify a bootstrap image to be loaded from a network server using TFTP or rcp.

You can enter the different types of boot commands in any order in the startup configuration file or in the BOOT environment variable. If you enter multiple boot commands, the DSLAM tries them in the order they are entered.

Loading from Flash Memory

Use this section to configure your DSLAM to boot from Flash memory. In the DSLAM, Flash memory is located in an internal SIMM. You can store or boot software images in Flash memory, as necessary. Flash memory can reduce the effects of network failure by reducing dependency on files that can be accessed only over the network.

Note Booting from ROM is faster than booting from Flash memory. However, if you are booting from a network server, Flash memory is faster and more reliable.

Flash Memory Tasks

Flash memory allows you to:

- Copy the system image to Flash memory using TFTP
- Copy the system image to Flash memory using rcp
- Copy a bootstrap image to Flash memory using TFTP or rcp
- Boot a DSLAM from Flash memory either automatically or manually
- Copy the Flash memory image to a network server using TFTP or rcp
- Copy the Flash memory bootstrap image to a network server using TFTP or rcp

Flash Memory Features

Flash memory features include:

- Flash memory can be remotely loaded with multiple system software images through TFTP or rcp transfers (one transfer for each file loaded).
- On the DSLAM, 8 MB of embedded Flash memory storage are provided.
- You can boot a DSLAM manually or automatically from a system software image stored in Flash memory, or you can boot from a network server using TFTP or rcp.

Security Precautions

Take these precautions when loading from Flash memory:

- Flash memory provides write protection against accidental erasing or reprogramming. You can remove the write-protect jumper, located next to the Flash components, to prevent reprogramming of embedded Flash memory.
- The system image stored in Flash memory can be changed only from the privileged EXEC level on the console terminal.

The DSLAM is shipped from the factory with the rxboot image in ROM. You can change the location of this image to embedded Flash memory. To specify the rxboot image Flash device, set the BOOTLDR environment variable.

Note When no BOOTLDR environment variable exists, the default rxboot image is the first image file in bootflash.

To configure the DSLAM for Flash memory:

Step 1 Set the BOOTLDR environment variable to change the location of the rxboot image that ROM uses for booting.

Step 2 Optionally, use rcp or TFTP to update the system image in embedded Flash memory. Performing this step allows you to update a degraded system image with one that is not degraded.

Step 3 Configure your system to automatically boot from the desired file in Flash memory. You may need to change the configuration register value. See the "Modifying the Configuration Register Boot Field" section on page 17-13 for more information on modifying the configuration register.

Step 4 Save your configurations.

Step 5 Power-cycle and reboot your system to ensure that the system is functioning properly.

Performing Flash Memory Configuration Tasks

Flash memory configuration tasks described in this section include configuring the DSLAM to automatically boot from an image in Flash memory. To configure a DSLAM to automatically boot from an image in Flash memory, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**boot system** [*filename*] DSLAM(config)#**boot system flash** [*filename*] DSLAM(config)#**boot system rcp** [*filename*] DSLAM(config)#**boot system mop** [*filename*] DSLAM(config)#**boot system tftp** [*filename*] DSLAM(config)#**boot system ftp** [*filename*]	Enter the filename of an image stored in Flash memory.
3.	DSLAM(config)#**config-register** *value*	Set the configuration register to enable loading of the system image from Flash memory.
4.	DSLAM(config)#**end**	Exit configuration mode.
5.	DSLAM#**copy running-config startup-config**	Save the configuration file to your startup configuration in the location specified by the CONFIG_FILE environment variable.
6.	DSLAM#**show startup-config**	Optionally, verify the contents of the startup configuration.
7.	DSLAM#**reload**	Power-cycle and reboot the system to ensure that the system is functioning properly.

If you enter more than one image filename, the DSLAM tries to recognize the filenames in the order entered. If a filename already appears in the configuration file and you want to specify a new filename, remove the existing filename by using the no boot system flash filename command.

Note The **no boot system** configuration command disables all boot system configuration commands regardless of the argument. If you specify the flash keyword or the filename argument using the **no boot system** command, this disables only the commands specified by these arguments.

Example

This example shows how to configure the DSLAM to automatically boot from an image in Flash memory:

```
DSLAM(config)#boot system flash 6260-wi-m_1.058.bin.Z
DSLAM(config)#config-register 0x1000
DSLAM(config)#end
DSLAM#copy running-config startup-config
[ok]
DSLAM#reload
[confirm] y

%SYS-5-RELOAD: Reload requested
booting /tftpboot/6260-wi-m_1.058.bin.Z 171.69.1.129
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Uncompressing file:  ###########################################################
################################################################################
################################################################################
################################################################################
################################################################################
################################################################################
######################################


Loading network-confg from 171.69.1.129 (via Ethernet0/0): !
[OK - 86/128975 bytes]

%SYS-5-CONFIG: Configured from network-confg by console tftp from 171.69.1.129
Loading /tftpboot/dslam-confg from 171.69.1.129 (via Ethernet0/0): !
[OK - 962/128975 bytes]

%SYS-4-CONFIG_NEWER: Configurations from version 11.1 may not be correctly understood.
%SYS-5-CONFIG: Configured from /tftpboot/dslam-confg by console tftp from 171.69.1.129
Loading 6260-wi-m_1.058.bin.Z from 171.69.1.129 (via Ethernet
0/0): ████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
██████████████████████████████████████
[OK - 2200823/7554184 bytes]

Uncompressing file:  ###########################################################
################################################################################
################################################################################
################################################################################
################################################################################
################################################################################
##############################################################################

<information deleted>

%SYS-5-RESTART: System restarted --

<information deleted>
```

After you have successfully configured Flash memory, you might want to configure the system with the **no boot system flash** command to revert to booting from ROM or bootflash. You might want to revert to booting from ROM or bootflash if you do not yet need this functionality, if you choose to boot from a network server, or if you do not have the proper image in Flash memory.

Loading from a Network Server

You can configure the DSLAM to load a system image from a network server using TFTP or rcp to copy the system image file.

To do so, you must set the configuration register boot field to the correct value. See the "Modifying the Configuration Register Boot Field" section on page 17-13.

If you do not boot from a network server using MOP and you do not specify either TFTP or rcp, by default, the system image that you specify is booted from a network server through TFTP.

Note If you are using a Sun workstation as a network server and TFTP to transfer the file, set up the workstation to enable verification and generation of User Datagram Protocol (UDP) checksums. See the Sun documentation for details.

For increased performance and reliability, use rcp to boot a system image from a network server. The rcp implementation uses the Transmission Control Protocol (TCP), which ensures reliable data delivery.

You cannot explicitly specify a remote username when you issue the boot command. Instead, the host name of the DSLAM is used. If the remote server has a directory structure, as do UNIX systems, and you boot the DSLAM from a network server using rcp, the DSLAM software searches for the system image on the server relative to the directory of the remote username.

You can also boot from a compressed image on a network server to ensure that there is enough memory available for storage.

If there is not enough room in memory to boot a regular image from a network server, you can create a compressed software image on any UNIX platform using the **compress** command. Refer to the documentation for your UNIX platform for the exact usage of the **compress** command.

To specify the loading of a system image from a network server, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**boot system** [**rcp** | **tftp**] *filename* [*ip-address*]	Specify the system image file to be booted from a network server using rcp or TFTP.
3.	DSLAM(config)#**config-register** *value*	Set the configuration register to enable loading of the system image from a network server.
4.	DSLAM(config)#**end**	Exit configuration mode.
5.	DSLAM#**copy running-config startup-config**	Save the configuration file to your startup configuration in the location specified by the CONFIG_FILE environment variable.

Example

In this example, the DSLAM uses rcp to boot from the "testme5.tester" system image file on a network server at IP address 131.108.0.1:

```
DSLAM(config)#boot system rcp testme5.tester 131.108.0.1
DSLAM(config)#config-register 0x010F
DSLAM(config)#end
DSLAM#copy running-config startup-config
```

Using a Fault-Tolerant Booting Strategy

Occasionally network failures make booting from a network server impossible. To lessen the effects of network failure, consider this booting strategy. After Flash is installed and configured, you might want to configure the DSLAM to boot in this order:

1. Boot an image from Flash.
2. Boot an image from a system file on a network server.
3. Boot from a ROM image.

This boot order provides the most fault-tolerant booting strategy. To allow the DSLAM to boot first from Flash, then from a system file from a network server, and finally from ROM, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**boot system** [*filename*] DSLAM(config)#**boot system flash:** [*filename*]	Configure the DSLAM to boot from Flash memory.
3.	DSLAM(config)#**boot system** [**rcp** | **mop** | **ftp** | **tftp**] *filename* [*ip-address*]	Configure the DSLAM to boot from a system filename.
4.	DSLAM(config)#**config-register** *value* [1]	Set the configuration register to enable loading of the system image from a network server or Flash.
5.	DSLAM(config)#**end**	Exit configuration mode.
6.	DSLAM#**copy running-config startup-config**	Save the configuration file to your startup configuration in the location specified by the CONFIG_FILE environment variable.

1. Refer to the "Modifying the Configuration Register Boot Field" section on page 17-13 for more information on systems that can use this command to modify the software configuration register.

Example

This example illustrates the order of the commands needed to implement this strategy. In the example, the DSLAM is configured to first boot an embedded Flash image called "gsxx". If that image fails, the DSLAM boots the configuration file "6260xx" from a network server.

```
DSLAM(config)#boot system flash 6260xx
DSLAM(config)#boot system 6260xx 131.131.101.101
DSLAM(config)#config-register 0x010F
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
```

```
DSLAM#copy running-config startup-config
[ok]
```

Using this strategy, a DSLAM has three sources from which to boot. These alternative sources help lessen the negative effects of a failure on the network or file server from which the system image is copied.

Specifying the Startup Configuration File

Configuration files can be stored on network servers. You can configure the DSLAM to automatically request and receive the following two configuration files from the network server at startup:

- Network configuration file
- Host configuration file

The server first attempts to load the network configuration file. This file contains information that is shared among several DSLAMs. For example, it can be used to provide mapping between IP addresses and host names.

The server next attempts to load the host configuration file. This file contains commands that apply to only one DSLAM. Both the network and host configuration files must be readable and must reside on a network server reachable using TFTP, rcp, or MOP.

You can specify an ordered list of network configuration and host configuration filenames. The DSLAM scans this list until it successfully loads the appropriate network or host configuration file.

In addition to storing configuration files on network servers with the DSLAM, you can store configuration files in NVRAM and in Flash memory. The CONFIG_FILE environment variable specifies the device and filename of the configuration file to use during initialization. For more information on environment variables, refer to the "Cisco Implementation of Environment Variables" section on page 17-39 in this chapter.

You can set the CONFIG_FILE environment variable to specify the startup configuration.

To specify a startup configuration file, perform *either* the first two tasks *or* the third task:

Step 1 Download the Network Configuration File

Step 2 Download the Host Configuration File

or

Download the CONFIG_FILE Environment Variable Configuration

Downloading the Network Configuration File

To configure the DSLAM to download a network configuration file from a server at startup, perform these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**boot network** [**tftp** | **rcp**] *filename* [*ip-address*]	Enter the network configuration filename to download a file using TFTP, rcp, or MOP.
3.	DSLAM(config)#**service config**[1]	Enable the DSLAM to automatically load the network file upon restart.
4.	DSLAM(config)#**end**	Exit configuration mode.
5.	DSLAM#**copy running-config startup-config**	Save the configuration file to your startup configuration in the location specified by the CONFIG_FILE environment variable.

1. For Step 2, if you do not specify a network configuration filename, the DSLAM uses the default filename "network-confg". If you omit the **tftp**, **rcp**, and **MOP** keywords, the DSLAM assumes that you are using TFTP to transfer the file and the server whose IP address you specify supports TFTP.

If you configure the DSLAM to download the network configuration file from a network server using rcp and the server has a directory structure (as do UNIX systems):

- The DSLAM software searches for the system image on the server relative to the directory of the remote username. The DSLAM host name is used as the remote username.
- You can specify more than one network configuration file. The DSLAM uses each file in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Downloading the Host Configuration File

To configure the DSLAM to download a host configuration file from a server at startup, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**boot host** [**tftp** | **rcp** | **mop**] *filename* [*ip-address*]	Optionally, enter the host configuration filename to be downloaded using rcp or TFTP.[1]
3.	DSLAM(config)#**service config**	Enable the DSLAM to automatically load the host file upon restart.
4.	DSLAM(config)#**end**	Exit configuration mode.
5.	DSLAM#**copy running-config startup-config**	Save the configuration file to your startup configuration in the location specified by the CONFIG_FILE environment variable.
6.	DSLAM#**reload**	Reset the DSLAM with the new configuration information.

1. If you do not specify a host configuration filename, the DSLAM uses its own name to form a host configuration filename by converting the DSLAM name to all lowercase letters, removing all domain information, and appending "-confg". If no host name information is available, the DSLAM uses the default host configuration filename dslam-confg.

You can specify more than one host configuration file. The DSLAM tries the files in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Example

In this example, the DSLAM is configured to boot from the host configuration file "hostfile1" and from the network configuration file "networkfile1":

```
DSLAM(config)#boot host hostfile1
DSLAM(config)#boot network networkfile1
DSLAM(config)#service config
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM#copy running-config startup-config
```

If the network server fails to load a configuration file during startup, it tries again every 10 minutes (the default) until a host provides the requested files. With each failed attempt, the network server displays a message on the console terminal. If the network server is unable to load the specified file, it displays the message:

```
Booting host-confg... [timed out]
```

The DSLAM uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the DSLAM detects a problem with NVRAM or the configuration it contains, the DSLAM enters the autoconfiguration mode. Refer to Chapter 3, "Initially Configuring the Cisco DSLAM" for more information on configuring the DSLAM.

Downloading the CONFIG_FILE Environment Variable Configuration

When you load startup configuration files from a server, you can configure the DSLAM to load a startup configuration file specified by the CONFIG_FILE environment variable. To do so, complete these tasks, beginning in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**copy** DSLAM#**copy flash** DSLAM#**copy running-config** DSLAM#**copy startup-config** DSLAM#**copy tftp**	Copy the configuration file to the device from which the DSLAM loads the file upon restart.
2.	DSLAM#**configure terminal**	Enter configuration mode from the terminal.
3.	DSLAM(config)#**boot config** *device*:*filename*	Set the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.
4.	DSLAM(config)#**end**	Exit configuration mode.
5	DSLAM#**copy running-config startup-config**	Save the runtime CONFIG_FILE environment variable to your startup configuration.
6	DSLAM#**show boot**	Optionally, verify the contents of the CONFIG_FILE environment variable.

When the DSLAM saves the runtime CONFIG_FILE environment variable to the startup configuration, the DSLAM saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable and saves a distilled version to NVRAM. The distilled version does not contain access list information. If NVRAM contains:

- A complete configuration file, the DSLAM prompts you to confirm the overwrite of the complete version with the distilled version.
- A distilled configuration file, the DSLAM does not prompt you for confirmation and overwrites the existing distilled configuration file in NVRAM.

Clearing the Configuration Information

To clear the contents of your startup configuration, use this command in privileged EXEC mode:

Command	Task
DSLAM#**erase startup-config**	Clear the contents of your startup configuration. This command erases the configuration specified by the CONFIG_FILE environment variable.

When you use the **erase startup-config** command, the DSLAM erases or deletes the configuration pointed to by the CONFIG_FILE environment variable. If this CONFIG_FILE environment variable specifies or points to:

- NVRAM, the DSLAM erases NVRAM.
- A Flash memory device and configuration filename, the DSLAM deletes the configuration file. That is, the DSLAM marks the file as "deleted," rather than erasing it. This feature allows you to recover a deleted file. Refer to the "Managing Flash Files" section on page 17-41 for more information on recovering deleted files.

To erase a saved configuration from a specific Flash device on a DSLAM, use one of these commands in privileged EXEC mode:

Command	Task
DSLAM#**erase** [*device*:] *filename* or DSLAM#**delete** [*device*:] *filename*	Erase or delete a specified configuration file on a specified Flash device. The DSLAM device must be bootflash.

As with the **erase startup-config** command, when you erase or delete a specific file, the system marks the file as deleted, allowing you to later recover a deleted file. If you omit the device, the DSLAM uses the default device specified by the **cd** command.

If you attempt to erase or delete the configuration file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to erase or delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

Examples

This example erases the "myconfig" file from embedded Flash:

```
DSLAM#erase nvram:myconfig
```

This example deletes the "myconfig" file from embedded Flash:

```
DSLAM#delete bootflash:myconfig
```

Storing System Images and Configuration Files

After modifying and saving your unique configurations, you can store them on a network server. You can use these network server copies of system images and configuration files as backup copies.

To store system images and configuration files, perform these tasks:

- Copy System Images from Flash Memory to a Network Server
- Copy Configuration Files from the DSLAM to a Network Server

Copying System Images from Flash Memory to a Network Server

You can copy system images from Flash memory to a TFTP server or an rcp server. You can use this server copy of the system image as a backup copy, or you can use it to verify that the copy in Flash is the same as the original file on disk. These sections describe these tasks:

- Copy from Flash Memory to a TFTP Server
- Copy from Flash Memory to an rcp Server

Copying from Flash Memory to a TFTP Server

You can copy a system image to a TFTP network server. In some implementations of TFTP, you must first create a "dummy" file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

To copy a system image to a TFTP network server, perform these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**show flash all** DSLAM#**show flash** [*device***:**]	(Optional) Display the name and note the exact spelling of the system image filename in Flash memory.
2.	DSLAM#**copy flash tftp** or DSLAM#**copy** *file_id* **tftp**	Copy the system image from Flash memory to a TFTP server.
3.	*ip-address* or *name*	At the prompt, enter the IP address or domain name of the TFTP server.
4.	*filename*	At the prompt, enter the filename of the system image in Flash memory.

Example

This example uses the **show flash all** command to learn the name of the system image file and the **copy flash tftp** command to copy the system image to a TFTP server. The name of the system image file appears in the filename listing at the top of the **show flash all** output.

```
DSLAM#show flash all
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. image    7B115AB2  8BC974   29  8898804 Oct 05 2000 01:09:14 ni2-dsl-mz.6
2   .D unknown  EE690AA0  8C7AFC   17    45320 Oct 05 2000 01:28:24 startup-cibe
3   .D unknown  2121A3AD  8D3E3C   17    49856 Oct 15 2000 03:41:26 startup-cibe
4   .. unknown  2121A3AD  8E017C   17    49856 Oct 18 2000 07:38:33 startup-cibe

6946436 bytes available (9044348 bytes used)

-------- F I L E   S Y S T E M   S T A T U S --------
  Device Number = 1
DEVICE INFO BLOCK: flash
  Magic Number          = 6887635   File System Vers = 10000    (1.0)
  Length                = 1000000   Sector Size      = 40000
  Programming Algorithm = 6         Erased State     = FFFFFFFF
  File System Offset    = 40000     Length = F40000
  MONLIB Offset         = 100       Length = C628
  Bad Sector Map Offset = 3FFF8     Length = 8
  Squeeze Log Offset    = F80000    Length = 40000
  Squeeze Buffer Offset = FC0000    Length = 40000
  Num Spare Sectors     = 0
    Spares:
STATUS INFO:
  Writable
  NO File Open for Write
  Complete Stats
  No Unrecovered Errors
  No Squeeze in progress
USAGE INFO:
  Bytes Used     = 8A017C  Bytes Available = 69FE84
  Bad Sectors    = 0       Spared Sectors  = 0
  OK Files       = 2       Bytes = 888BB4
  Deleted Files  = 2       Bytes = 173C8
  Files w/Errors = 0       Bytes = 0
```

A series of Cs indicates that a checksum verification of the image is occurring, and an exclamation point indicates that the copy process is occurring. To stop the copy process, press **Ctrl**-**^**.

This example uses the **show flash** [*device*:] command to display the name of the system image file to copy.

The file to copy is “test”. The example uses the **copy** *file_id* **tftp** command to copy “test” to a TFTP server.

```
DSLAM#show flash slot0:
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. FFFFFFFF 129EECA3 214D4   13   5204     May 03 1996 14:07:35 backup-config
2   .. 1        AE9B32B  22A68   14   5393     May 03 1996 15:32:57 startup-config
3   .. FFFFFFFF E9D05582 247730  23   2247751  May 04 1996 12:08:51 6260-wi-m_1.1(1)
4   .. FFFFFFFF E9D05582 46C3F8  4    2247751  May 04 1996 13:25:14 test

3488776 bytes available (4506616 bytes used)
DSLAM#copy bootflash:test tftp
Enter destination file name [test]:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Address or name of remote host [dirt.cisco.com]? 171.69.1.129
!
```

A series of Cs indicates that a checksum verification of the image is occurring, and an exclamation point indicates that the copy process is occurring.

After you configure Flash memory, you might want to configure the system (using the **configure terminal** command) with the **no boot system flash** configuration command to revert to booting from ROM. For example, you might want to revert to booting from ROM if you do not yet need this functionality, if you choose to boot from a network server, or if you do not have the proper image in Flash memory. After you enter the **no boot system flash** command, use the **copy running-config startup-config** command to save the new configuration command to the startup configuration.

This procedure on the DSLAM also requires changing the processor's configuration register. Refer to the "Modifying the Configuration Register Boot Field" section on page 17-13 for instructions.

Copying from Flash Memory to an rcp Server

You can copy a system image from Flash memory to an rcp network server.

The rcp protocol requires a client to send the remote username on each rcp request to the server. When you copy an image from Flash memory to a network server using rcp, the DSLAM software sends the remote username associated with the current TTY (terminal) process, if that name is valid. If the TTY remote username is invalid, the DSLAM software uses the DSLAM host name as both the remote and local user names.

Note For Cisco, TTYs are commonly used in communication servers. The concept of TTY originated with UNIX. For UNIX systems, each physical device is represented in the file system. Terminals are called "TTY devices", which stands for" teletype", the original UNIX terminal.

You can configure a different remote username to be sent to the server. If the network server has a directory structure, as do UNIX systems, the rcp protocol implementation writes the system image to the directory associated with the remote username on the network server.

For the rcp command to execute properly, an account must be defined on the destination server for the remote username.

To stop the copy process, press **Ctrl**-^.

If you copy the system image to a personal computer used as a file server, the computer must support the rcp protocol.

To copy the system image from Flash memory to a network server, perform these tasks, beginning in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**show flash all** DSLAM#**show flash** [*device:*]	(Optional) If you do not already know it, learn the exact spelling of the system image filename in Flash memory. On the DSLAM, you can learn the spelling of the system image filename in embedded Flash memory.
2.	DSLAM#**configure terminal**	Enter configuration mode from the terminal. This step is required only if you are going to override the default remote username in the next step.

Step	Command	Task
3.	DSLAM(config)#**ip rcmd remote-username** *username*	Specify the remote username. This step is optional, but recommended.
4.	DSLAM(config)#**end**	Exit configuration mode.
5.	DSLAM#**copy flash rcp** DSLAM#**copy** *file_id* **rcp**	Using rcp, copy the system image in Flash memory to a network server.
6.	*ip-address* or *name*	When prompted, enter the IP address or domain name of the rcp server.
7.	*filename*	When prompted, enter the filename of the system image in Flash memory.

Examples

This example shows how to copy the system image file from Flash memory to a network server using rcp:

```
DSLAM#configure terminal
DSLAM(config)#ip rcmd remote-username netadmin2
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM#copy flash rcp
Enter source file name: 6260-wi-m_1.1(1)
Enter destination file name [6260-wi-m_1.1(1)]:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Address or name of remote host [dirt.cisco.com]? 171.69.1.129
Writing 6260-wi-m_1.1(1) ################################################
################################################################################
################################################################################
################################################################################
################################################################################
########################################################################
```

This example shows how to copy a system image file from embedded Flash to a network server using rcp:

```
DSLAM#configure terminal
DSLAM(config)#ip rcmd remote-username netadmin2
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM#copy bootflash:6260-wi-m_1.1(1) rcp
Enter destination file name [6260-wi-m_1.1(1)]:
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Address or name of remote host []? 171.69.1.129
Writing 6260-wi-m_1.1(1) ################################################
################################################################################
################################################################################
################################################################################
################################################################################
########################################################################
```

The screen filled with exclamation points indicates that the process is working.

Copying Configuration Files from the DSLAM to a Network Server

You can copy configuration files from the DSLAM to a TFTP server or rcp server. You might do this task to back up a current configuration file to a server before changing its contents, thereby allowing you to later restore the original configuration file from the server. These sections describe these tasks:

- Copy from the DSLAM to a TFTP Server
- Copy from the DSLAM to an rcp Server

Copying from the DSLAM to a TFTP Server

Usually, the configuration file that you copy to must already exist on the TFTP server and be globally writable before the TFTP server allows you to write to it.

To store configuration information on a TFTP network server, complete these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**copy running-config tftp** or DSLAM#**copy startup-config tftp**	Specify that the running or startup configuration file will be stored on a network server.
2.	*ip-address*	Enter the IP address of the network server.
3.	*filename*	Enter the name of the configuration file to store on the server.
4.	**y**	Confirm the entry.

Example

This example shows how to copy a running configuration file from a DSLAM to a TFTP server:

```
DSLAM#copy running-config tftp
Remote host []? 171.69.1.129
Name of configuration file to write [dslam-confg]? backup-confg
Write file backup-confg on host 171.69.1.129? [confirm] y
Building configuration...

Writing backup-confg !!! [OK]
```

Copying from the DSLAM to an rcp Server

You can use rcp to copy configuration files from the local DSLAM to a network server. You can copy a running configuration file or a startup configuration file to the server.

The rcp protocol requires that a client send the remote username on each rcp request to a server. When you issue a command to copy a configuration file from the DSLAM to a server using rcp, the DSLAM sends a default remote username unless you override the default by configuring a remote username. By default, the DSLAM software sends the remote username associated with the current TTY (terminal) process, if that name is valid.

If the TTY remote username is invalid, the DSLAM software uses the DSLAM host name as both the remote and local user names. If the server has a directory structure, as do UNIX systems, the rcp protocol implementation writes the configuration file to the directory associated with the remote username on the server.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username.

If you copy the configuration file to a personal computer used as a file server, the computer must support rcp.

To copy a startup configuration file or a running configuration file from the DSLAM to an rcp server, perform one of following tasks:

- Copy a Running Configuration File to an rcp Server
- Copy a Startup Configuration File to an rcp Server

Copy a Running Configuration File to an rcp Server

You can copy the running configuration file to an rcp server. The copied file can serve as a backup configuration file.

To store a running configuration file on a server, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**ip rcmd remote-username** *username*	Specify the remote username. This step is optional, but recommended.
3.	DSLAM(config)#**end**	Exit from global configuration mode.
4.	DSLAM#**copy running-config rcp**	Specify that the DSLAM's running configuration file will be stored on a network server.
5.	*ip-address*	Enter the IP address of the network server.
6.	*filename*	Enter the name of the configuration file to store on the server.
7.	**y**	Confirm the entry.

Example

This example shows how to copy the running configuration file named "dslam-confg" to the "netadmin1" directory on the remote host with an IP address of 171.69.1.129:

```
DSLAM(config)#ip rcmd remote-username netadmin2
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM#copy running-config rcp
Remote host []? 171.69.1.129
Name of configuration file to write [dslam-confg]?
Write file dslam-confg on host 171.69.1.129? [confirm] y
Building configuration...

Writing dslam-confg !! [OK]
```

Copying a Startup Configuration File to an rcp Server

You can copy the contents of the startup configuration file to an rcp server. The copied file can serve as a backup configuration file.

To copy a startup configuration file to a network server using rcp, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**ip rcmd remote-username** *username*	Specify the remote username. This step is optional, but recommended.
3.	DSLAM(config)#**end**	Exit from global configuration mode.
4.	DSLAM#**copy startup-config rcp**	Copy the configuration file specified by the CONFIG_FILE environment variable to an rcp server.
5.	*ip-address*	Enter the IP address of the network server.
6.	*filename*	Enter the name of the configuration file to store on the server.
7.	**y**	Confirm the entry.

Example

This example shows how to store a startup configuration file on a server by using rcp to copy the file:

```
DSLAM#configure terminal
DSLAM(config)#ip rcmd remote-username netadmin2
DSLAM(config)#end
DSLAM#
%SYS-5-CONFIG_I: Configured from console by console
DSLAM#copy startup-config rcp
Remote host []? 171.69.1.129
Name of configuration file to write [dslam-confg]?
Write file dslam-confg on host 171.69.1.129? [confirm] y
Writing dslam-confg !! [OK]
```

Configuring a DSLAM as a TFTP Server

It is both costly and inefficient to have a dedicated TFTP server on every network segment. To cut costs and time delays in your network, you can configure a DSLAM as a TFTP server.

Typically, the DSLAM configured as a server forwards operating system images from its Flash memory to other DSLAMs. You can also configure the DSLAM to respond to other types of service requests, such as Reverse Address Resolution Protocol (RARP) requests.

To configure the DSLAM as a server, perform any of these tasks. The tasks are not mutually exclusive.

- Designate a DSLAM as a TFTP Server
- Configure Flash Memory as a TFTP Server

Designating a DSLAM as a TFTP Server

As a TFTP server host, the DSLAM responds to TFTP Read Request messages by sending a copy of the system image contained in ROM or one of the system images contained in Flash memory to the requesting host. The TFTP Read Request message must use one of the filenames specified in the DSLAM's configuration.

To specify TFTP server operation for a DSLAM, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**tftp-server rom alias** *filename1* [*access-list-number*] DSLAM(config)#**tftp-server flash** *device:filename*	Specify TFTP server operation.
3.	DSLAM(config)#**end**	Exit configuration mode.
4.	DSLAM#**copy running-config startup-config**	Save the running configuration file to the startup configuration location specified by the CONFIG_FILE environment variable.

The TFTP session can sometimes fail. TFTP generates these special characters to help you determine why a TFTP session failed:

- An "E" character indicates that the TFTP server received an erroneous packet.
- An "O" character indicates that the TFTP server received an out-of-sequence packet.
- A period (.) indicates a timeout.

The transfer session might still succeed if TFTP generates these characters, but the output is useful for diagnosing the transfer failure.

Examples

In this example, the system uses TFTP to send a copy of the Flash memory file *version-1.03* in response to a TFTP Read Request for that file. The requesting host is checked against access list 22.

```
DSLAM(config)#tftp-server flash version-1.03 22
```

In this example, the system uses TFTP to send a copy of the ROM image *6260-m_1.101* in response to a TFTP Read Request for the *6260-m_1.101* file:

```
DSLAM(config)#tftp-server rom alias 6260-m_1.101
```

Configuring Flash Memory as a TFTP Server

Flash memory can be used as a TFTP file server for other DSLAMs on the network. This feature allows you to boot a remote DSLAM with an image that resides in the Flash server memory.

The DSLAM allows you to specify one of the different Flash memory devices as the TFTP server. You must specify embedded Flash (bootflash:) as the TFTP server.

In the description that follows, one DSLAM is referred to as the *Flash server*, and all other DSLAMs are referred to as *client DSLAMs*. Example configurations for the Flash server and client DSLAMs include commands, as necessary.

To configure Flash memory as a TFTP server, perform these tasks:

- Perform Prerequisite Tasks
- Configure the Flash Server
- Configure the Client DSLAM

Performing Prerequisite Tasks

The Flash server and client DSLAM must be able to reach each other before the TFTP function can be implemented. Verify this connection by pinging between the Flash server and the client DSLAM (in either direction) with the **ping** command.

An example of the **ping** command follows:

```
DSLAM#ping 131.152.1.129

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.152.1.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

In this example, the IP address of 131.152.1.129 belongs to the client DSLAM. Connectivity is indicated by a series of exclamation points (!), while a series of periods (.) plus "*timed out" or "failed"* indicates no connection. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and the client DSLAM, and ping again.

After you verify the connection, ensure that a TFTP-bootable image is present in Flash memory. This is the system software image the client DSLAM boots. Note the name of this software image so you can verify it after the first client boot.

Note The filename used must represent a software image that is present in Flash memory. If no image resides in Flash memory, the client DSLAM boots the server's ROM image by default.

Caution For full functionality, the software residing in the Flash memory must be the same type as the ROM software installed on the client DSLAM. For example, if the server has X.25 software and the client does not have X.25 software in ROM, the client will not have X.25 capabilities after booting from the server's Flash memory.

Configuring the Flash Server

To configure the Flash server, use this command in global configuration mode:

Command	Task
DSLAM(config)#**tftp-server flash** *device*:*filename*	Specify the TFTP server operation for a DSLAM.

Example

This example shows how to configure the Flash server. This example gives the filename of the software image in the Flash server and one access list (labeled "*1*"). The access list must include the network where the client DSLAM resides. Thus, in the example, the network 131.108.101.0 and any client DSLAMs on it can access the Flash server file 6260-m_1.9.17.

```
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U; end with CTRL/Z
Server(config)#tftp-server flash 6260-m_1.9.17 1
Server(config)#access-list 1 permit 131.108.101.0 0.0.0.255
Server(config)#end
Server#copy running-config startup-config
[ok]
```

Configuring the Client DSLAM

You can configure the client DSLAM to first load a system image from the Flash server, then, as a backup, configure the client DSLAM to then load its own ROM image if the load from a Flash server fails. To do so, complete these tasks, beginning in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**no boot system**	Remove all previous **boot system** statements from the configuration file.
3.	DSLAM(config)#**boot system** [**rcp** | **tftp**] *filename* [*ip-address*]	Specify that the client DSLAM loads a system image from the Flash server.
4.	DSLAM(config)#**config-register** *value*	Set the configuration register to enable the client DSLAM to load a system image from a network server.
5.	DSLAM(config)#**end**	Exit configuration mode.
6.	DSLAM#**copy running-config startup-config**	Save the running configuration file to the startup configuration location specified by the CONFIG_FILE environment variable.
7.	DSLAM#**reload**	Reload the DSLAM to make your changes take effect.

Caution Using the no boot system command, as in this example, will invalidate *all* other boot system commands currently in the client DSLAM system configuration. Before proceeding, determine whether or not the system configuration stored in the client DSLAM first requires saving (uploading) to a TFTP file server so that you have a backup copy.

Example

This example shows how to use these commands:

```
Client(config)#no boot system
Client(config)#boot system 6260-m_1.9.17 131.131.111.111
Client(config)#boot system rom
Client(config)#config-register 0x010F
```

```
Client(config)#end
Client#copy running-config startup-config
[ok]
Server#reload
```

In this example, the no boot system command invalidates all other boot system commands currently in the configuration memory, and any boot system command entered after this command is executed first. The second command, **boot system** *filename address*, tells the client DSLAM to look for the file *6260-m_1.9.17* in the (Flash) server with an IP address of 131.131.111.111. Failing this, the client DSLAM boots from its system ROM in response to the **boot system rom** command, which is included as a backup in case of a network problem. The **copy running-config startup-config** command copies the configuration to NVRAM to the location specified by the CONFIG_FILE environment variable, and the **reload** command boots the system.

Caution The system software (*6260-m_1.9.17* in the example) to be booted from the Flash server (131.131.111.111 in the example) must reside in Flash memory on the server. If it is not in Flash memory, the client DSLAM boots the Flash server's system ROM.

Verifying the Client DSLAM

To verify that the software image booted from the Flash server is the image in Flash memory, use the EXEC command.

Command	Task
DSLAM#**show version**	Verify that the software image booted from the Flash server is the image present in Flash memory of the client DSLAM.

This example shows output of the **show version** command:

```
DSLAM#show version
Cisco Internetwork Operating System Software
IOS (tm) PNNI Software (6260-WP-M), Version XX.X(X), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1998 by cisco Systems, Inc.
Compiled Tue 07-Oct-97 04:53 by
Image text-base: 0x60010910, data-base: 0x604E6000

ROM: System Bootstrap, Version XX.X(X.X.WAX.0) [integ 1.4.WAX.0], RELEASE SOFTWARE

DSLAM uptime is 2 weeks, 2 days, 39 minutes
System restarted by power-on
System image file is "bootflash:6260-wp-mz.112-8.0.1.FWA4.0.16", booted via bootflash

cisco ASP (R4600) processor with 65536K bytes of memory.
R4700 processor, Implementation 33, Revision 1.0
Last reset from power-on
1 Ethernet/IEEE 802.3 interface(s)
20 ATM network interface(s)
123K bytes of non-volatile configuration memory.

8192K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x2101
```

The important information in this example is contained in the second line "IOS (tm)...," which shows the version of the operating system in the client DSLAM's RAM. The second "ROM: ...." line shows the filename of the system image loaded from the Flash server.

Note If no bootable image is present in the Flash server memory when the client server is booted, the version currently running (the first line of the **show version** output) is the system ROM version of the Flash server by default.

Verify that the software shown in the first line of the **show version** output is the software residing in the Flash server memory.

Configuring the DSLAM for Other Types of Servers

You can configure the DSLAM to work with various types of servers. Specifically, you can configure the DSLAM to forward different types of service requests.

Specifying Asynchronous Interface Extended BOOTP Requests

The Boot Protocol (BOOTP) server for asynchronous interfaces supports the extended BOOTP requests specified in RFC 1084. This command is helpful in conjunction with using the auxiliary port as an asynchronous interface.

To configure extended BOOTP requests for asynchronous interfaces, use this command in global configuration mode:

Command	Task
`DSLAM(config)#`**`async-bootp`** *tag* `[:`*hostname*`]` *data*	Configure extended BOOTP requests for asynchronous interfaces.

To display the extended BOOTP requests, use this privileged EXEC command:

Command	Task
`DSLAM#`**`show async bootp`**	Show parameters for BOOTP requests.

Performing Optional Startup Tasks

This sections describe optional startup tasks:

- Copy a File into a Flash Partition
- Configure the DSLAM to Automatically Boot from embedded Flash Memory
- Additional DSLAM Functions

Copying a File into a Flash Partition

To download a file into a Flash, use one of these commands in privileged EXEC mode:

Command	Task
DSLAM#**copy tftp flash**	Download a file from a TFTP server into a Flash partition.
DSLAM#**copy rcp flash**	Download a file from an rcp server into a Flash partition.

Configuring the DSLAM to Automatically Boot from Embedded Flash Memory

To configure the DSLAM to boot automatically from embedded Flash, use this command in global configuration mode:

Command	Task
DSLAM(config)#**boot system flash** *filename*	Boot the specified file from the first partition.

The result of booting a relocatable image from Flash depends on where and how the image was downloaded into Flash memory. The following describes the various ways an image might be downloaded and the corresponding results of booting from Flash memory.

Method of Downloading	Result of Booting from Flash
The image was downloaded as the first file by a nonrelocatable image.	The image executes in place from Flash memory, like a run-from-Flash image.
The image was not downloaded as the first file by a nonrelocatable image.	The nonrelocatable image will not relocate the image before storage in Flash memory. This image will not be booted.
The image was downloaded as the first file by a relocatable image.	The image executes in place from Flash memory, like a run-from-Flash image.
The image was not downloaded as the first file by a relocatable image.	The relocatable image relocates the image before storage in Flash memory. Hence, the image executes in place from Flash memory, like any other run-from-Flash image.

Additional DSLAM Functions

These sections describe additional DSLAM functions:

- Copy a Boot Image
- Verify a Boot Image Checksum
- Erase Boot Flash Memory

Copying a Boot Image

You can copy a boot image from an rcp, TFTP, or MOP server to boot Flash memory. You can also copy the boot image from the boot Flash memory to an rcp or TFTP server by using one of these commands in privileged EXEC mode:

Command	Task
DSLAM#**copy tftp bootflash** or DSLAM#**copy rcp bootflash**	Copy a boot image from an TFTP or rcp server to boot Flash memory.

To copy a boot image from boot Flash memory to an rcp or TFTP server, perform this task in privileged EXEC mode:

Command	Task
DSLAM#**copy bootflash** {**rcp** | **tftp**}	Copy a boot image from boot Flash memory to an rcp or TFTP server.

Verifying a Boot Image Checksum

To verify the checksum of a boot image in boot Flash memory, use the EXEC command:

Command	Task
DSLAM>**verify bootflash:**	Verify the checksum of a boot image.

Erasing Boot Flash Memory

To erase the contents of boot Flash memory, use this command in privileged EXEC mode:

Command	Task
DSLAM# **erase bootflash:**	Erase boot Flash memory.

Performing DSLAM Startup Tasks

This section describes Cisco's implementation of environment variables on the DSLAM. It also describes startup tasks in these sections:

- Format Flash Memory
- Manage Flash Files
- Load and Display Software Images Over the Network

Cisco Implementation of Environment Variables

Embedded Flash memory can store executable images and configuration files. The DSLAM can now boot images and load configuration files from embedded Flash, NVRAM, and the network.

Because the DSLAM can boot images and load configuration files from several locations, these systems use special ROM monitor environment variables to specify the location and filename of images and configuration files that the DSLAM uses for various functions. These special environment variables are:

- BOOT Environment Variable
- BOOTLDR Environment Variable
- CONFIG_FILE Environment Variable
- Control Environment Variables

BOOT Environment Variable

The BOOT environment variable specifies a list of bootable images on various devices. The only valid device is embedded Flash (bootflash:). Once you save the BOOT environment variable to your startup configuration, the DSLAM checks the variable upon startup to determine the device and filename of the image to boot.

The DSLAM tries to boot the first image in the BOOT environment variable list. If the DSLAM cannot boot that image, it tries to boot the next image specified in the list. The DSLAM tries each image in the list until it successfully boots. If the DSLAM cannot boot any image in the BOOT environment variable list, it attempts to boot the ROM image.

If an entry in the BOOT environment variable list does not specify a device, the DSLAM assumes the device is tftp. If an entry in the BOOT environment variable list specifies an invalid device, the DSLAM skips that entry.

BOOTLDR Environment Variable

The BOOTLDR environment specifies the Flash device and filename containing the rxboot image that the ROM monitor uses. The only valid device is bootflash:.

This environment variable allows you to have several rxboot images. You can also instruct the ROM monitor to use a specific rxboot image without having to DSLAM out ROMs. After you save the BOOTLDR environment variable to your startup configuration, the DSLAM checks the variable upon startup to determine which rxboot image to use.

CONFIG_FILE Environment Variable

The CONFIG_FILE environment variable specifies the device and filename of the configuration file to use for initialization (startup). The only valid device is embedded Flash (bootflash:). After you save the CONFIG_FILE environment variable to your startup configuration, the DSLAM checks the variable upon startup to determine the location and filename of the configuration file to use for initialization.

The DSLAM uses the NVRAM configuration during initialization when the CONFIG_FILE environment variable does not exist or when it is null (such as at first-time startup). If the DSLAM detects a problem with NVRAM or the configuration it contains, the DSLAM enters the autoconfiguration mode. Refer to the Chapter 3, “Initially Configuring the Cisco DSLAM.”

Control Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain system image commands. To create or modify the BOOT, BOOTLDR, and CONFIG_FILE environment variables, use the **boot system**, **boot bootldr**, and **boot config** system image commands, respectively.

Note When you use these three global configuration commands, you affect only the running configuration. You must save the environment variable settings to your startup configuration to put the information under ROM monitor control and for the environment variables to function as expected. Use the **copy running-config startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT, BOOTLDR, and the CONFIG_FILE environment variables by issuing the **show boot** command. This command displays the settings for these variables as they exist in the startup configuration and in the running configuration if a running configuration setting differs from a startup configuration setting.

Use the **show startup-config** command to display the contents of the configuration file pointed to by the CONFIG_FILE environment variable.

Formatting Flash Memory

You must format embedded Flash memory before using it.

You can reserve certain Flash memory sectors as spares for use when other sectors fail. Use the **format** command to specify between 0 and 16 sectors as spares. If you reserve a small number of spare sectors for emergencies, you do not waste space because you can use most of Flash memory. If you specify zero spare sectors and some sectors fail, you must reformat Flash memory and erase all existing data.

The system requires a monlib file for the format operation. The monlib file is the ROM monitor library. The ROM monitor uses the monlib file to access files in the Flash file system. The system software contains the monlib file.

Caution The formatting procedure erases all information in Flash memory. To prevent the loss of important data, proceed carefully.

To format Flash memory, use this command in privileged EXEC mode:

Command	Task
DSLAM#**format** [**spare** *spare-number*] *device1*: [[*device2*:][*monlib-filename*]]	Format Flash memory.

Example

This example shows how to use the **format** command to format embedded Flash memory:

```
DSLAM#format bootflash:
Running config file on this device, proceed? [confirm] y
All sectors will be erased, proceed? [confirm] y
Enter volume id (up to 31 characters):
```

```
Formatting sector 1 (erasing)
Format device slot0 completed
```

When the DSLAM returns you to the EXEC prompt, Flash memory is successfully formatted and ready for use.

Recovering from Locked Blocks

You can also format Flash memory to recover from locked blocks. A locked block of Flash memory occurs when power is lost during a write or erase operation. When a block of Flash memory is locked, it cannot be written to or erased, and the operation will consistently fail at a particular block location. The only way to recover from locked blocks is by reformatting Flash memory with the **format** command.

Caution Formatting Flash memory to recover from locked blocks will cause existing data to be lost.

Managing Flash Files

You can manage files on embedded flash memory. These sections describe the tasks you help you manage your files:

- Set the System Default Flash Device (always **bootflash** for the DSLAM)
- Display the Current Default Flash Device
- Show a List of Files on a Flash Device
- Delete Files on a Flash Device

Setting the System Default Flash Device

You can specify the Flash device that the system uses as the default device. Setting the default Flash device allows you to omit an optional *device***:** argument from related commands. For all EXEC commands that have an optional *device***:** argument, the system uses the device specified by the **cd** command when you omit the optional *device***:** argument. For example, the **dir** command contains an optional *device***:** argument and displays a list of files on a Flash memory device.

DSLAM requires that the Flash device be **bootflash**, for embedded Flash. Setting **bootflash** as the default lets you skip the *device*: parameter.

To specify a default Flash device, use this command in EXEC mode:

Command	Task
DSLAM>**cd** *device*:	Set a default Flash memory device.

Example

This example shows how to set the default device to embedded Flash (the only option for DSLAM):

```
DSLAM>cd bootflash:
```

Displaying the Current Default Flash Device

You may want to show the current setting of the **cd** command to see which device is the current default Flash device. To display the current default Flash device specified by the **cd** command, use this command in EXEC mode:

Command	Task
DSLAM>**pwd**	Display the current Flash memory device.

Examples

This example shows that the present working device specified by the **cd** command is bootflash:

```
DSLAM>pwd
bootflash
```

This example shows how to use the **cd** command to change the present working device to bootflash and then uses the **pwd** command to display that present working device:

```
DSLAM>cd bootflash:
DSLAM>pwd
bootflash:/
```

Showing a List of Files in Embedded Flash

You may want to view a list of the contents of embedded Flash before manipulating its contents. For example, before copying a new configuration file to Flash, you may want to verify that the device does not already contain a configuration file with the same name. Similarly, before copying a Flash configuration file to another location, you may want to verify its filename for use in another command. You can check the contents of embedded flash with the **dir** EXEC command.

To show a list of files on a specified Flash device, use the EXEC command:

Command	Task
DSLAM>**dir** [**/all**] bootflash: [*filename*]	Display a list of files in embedded Flash.

Examples

This example shows how to instruct the DSLAM to list undeleted files for the default device specified by the **cd** command. Notice that the DSLAM displays the information in short format because no keywords are used:

```
Directory of bootflash:/

    1  -rw-     3419352   Sep 26 2000 23:59:56  ni2-dboot-mz.121-6.DA

3801088 bytes total (381608 bytes free)
```

This example shows how to display the long version of the same device:

```
DSLAM#dir /long
-#- ED --type-- --crc--- -seek-- nlen -length- -----date/time------ name
1   .. config   217B75D1   20E04   14     3458 Sep 29 1997 17:36:02 startup-config
2   .. unknown  2F9F6B8B  2CF9C0   29  2812732 Nov 11 1997 14:23:43 6260-wp-mz.113-0.8.TWA4.1.1
```

```
5178944 bytes available (2816448 bytes used)
```

Deleting Files in Embedded Flash

When you no longer need a file in Flash, you can delete it.

Caution Be careful not to delete your only known good boot image. If you have enough available Flash memory, create a backup image. The backup image allows you to revert to a known good boot image if you have trouble with the new image. If you delete all boot images you can no longer download any images.

To delete a file from embedded Flash, use one of these commands in privileged EXEC mode:

Command	Task
DSLAM#**delete bootflash:***filename* or DSLAM#**erase nvram:***filename*	Delete a file from embedded Flash.

If you attempt to delete the configuration file specified by the CONFIG_FILE or BOOTLDR environment variable, the system prompts you to confirm the deletion. Also, if you attempt to delete the last valid system image specified in the BOOT environment variable, the system prompts you to confirm the deletion.

Examples

This example shows how to delete the “myconfig” file from embedded Flash:

```
DSLAM#delete bootflash:myconfig
```

This example shows how to erase the “myconfig” file from embedded Flash:

```
DSLAM#erase nvram:myconfig
```

Loading and Displaying Software Images Over the Network

Each ASP has a writable control store (WCS) that stores software. You can load updated software onto the WCS from the on-board ROM or from Flash memory.

With this feature, you can update software without having physical access to the DSLAM, and you can load new software without rebooting the system.

To load software from Flash memory, complete these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**copy tftp flash** or DSLAM#**copy tftp** *file_id*	Copy software files into Flash. See the "Copying System Software Images from a Network Server to the DSLAM" section on page 17-2 for more information about how to copy TFTP images to Flash memory.
2.	DSLAM#**copy running-config startup-config**	Retain new configuration information when the system is rebooted.

If an error occurs when you are attempting to download software, the system loads the default system software image. The default software image is bundled with the system software.

These configuration commands are implemented after one of these three events:

- The system is booted.
- A card is inserted or removed.
- The configuration command **reload** is issued.

After you have entered a software configuration command and one of these events has taken place, all cards are reset, loaded with software from the appropriate sources, tested, and enabled for operation.

To signal to the system that all software configuration commands have been entered and the processor cards should be reloaded, use this command in privileged EXEC mode:

Command	Task
DSLAM#**reload**	Notify the system that all software configuration commands have been entered and the processor cards should be reloaded.

If Flash memory is busy, or a **software reload** command is executed while Flash is locked, the files will not be available and the on-board ROM software will be loaded. Issue another **software reload** command when Flash memory is available to load the proper software. The **show flash** command will show if another user or process has locked Flash memory.

The **software reload** command should not be used while Flash is in use. For example, do not use this command when a **copy tftp flash** or **show flash** command is active.

The **software reload** command is automatically added to your running configuration when you issue a software command that changes the system's default behavior.

Configuring the Remote Shell and Remote Copy Functions

You can optionally configure your DSLAM for remote shell (rsh) and rcp functions. This feature allows you to execute commands on remote DSLAMs and to remotely copy system images and configuration files to and from a network server or a DSLAM.

This section provides a description of the Cisco implementation of rsh and rcp and describes the tasks to configure the DSLAM for rsh and rcp in these subsections:

- Configure a DSLAM to Support Incoming rcp Requests and rsh Commands

- Configure the Remote Username for rcp Requests
- Remotely Execute Commands Using rsh

Cisco Implementation of rsh and rcp Protocols

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the rsh protocol, which included the rsh and rcp functions. Rsh and rcp give you the ability to execute commands remotely and copy files to and from a file system residing on a remote host or server on the network. Cisco's implementation of rsh and rcp interoperates with standard implementations.

Using the rsh Protocol

From the DSLAM, you can use rsh protocol to execute commands on remote systems to which you have access. When you issue the rsh command, a shell is started on the remote system. The shell allows you to execute commands on the remote system without having to log in to the target host.

You do not need to connect to the system or DSLAM and then disconnect after you execute a command when using rsh. For example, you can use rsh to remotely look at the status of other DSLAMs without connecting to the target DSLAM, executing the command, and then disconnecting from the DSLAM. This is useful for looking at statistics on many different DSLAMs.

Maintaining rsh Security

To gain access to a remote system running rsh, such as a UNIX host, there must be an entry in the system's .rhosts file or its equivalent to identify you as a trusted user who is authorized to execute commands remotely on the system. On UNIX systems, the .rhosts file identifies trusted users who can remotely execute commands on the system.

You can enable rsh support on a Cisco DSLAM to allow users on remote systems to execute commands on the DSLAM. However, the Cisco implementation of rsh does not support an .rhosts file. Instead, you configure a local authentication database to control access to the DSLAM by users attempting to execute commands remotely using rsh. A local authentication database is similar in concept and use to a UNIX .rhosts file. Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user.

Using the rcp Protocol

The rcp copy commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you do not need to create a server for file distribution, as you do with TFTP. You only need to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission on the destination file. If the destination file does not exist, rcp creates it for you.

Although the Cisco rcp implementation emulates the behavior of the UNIX rcp implementation—copying files among systems on the network—the command syntax differs from the UNIX rcp command syntax. Cisco rcp support offers a set of copy commands that use rcp as the transport mechanism. These rcp copy commands are similar to the Cisco TFTP copy commands, but they offer faster performance and reliable delivery of data. These improvements are possible because the rcp

transport mechanism is built on and uses the TCP/IP stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the DSLAM to a network server, and vice versa.

You can also enable rcp support on the DSLAM to allow users on remote systems to copy files to and from the DSLAM.

Configuring a DSLAM to Support Incoming rcp Requests and rsh Commands

You configure a local authentication database to control access to the DSLAM by remote users. To allow remote users to execute rcp or rsh commands on the DSLAM, configure entries for those users in the authentication database of the DSLAM.

Each entry configured in the authentication database identifies the local user, the remote host, and the remote user. You can specify the DSLAM host name as the local username. To be allowed to remotely execute commands on the DSLAM, the remote user must specify all three values—the local username, the remote host name, and the remote username—and must be able to identify the local username. For rsh users, you can also grant a user permission to execute privileged EXEC commands remotely.

To make the local username available to remote users, you must communicate the username to the network administrator or the remote user. To allow a remote user to execute a command on the DSLAM, Cisco's rcp implementation requires that the local username sent by the remote user match the local username configured in the database entry.

The DSLAM software uses Domain Name System (DNS) to authenticate the remote host's name and address. Because DNS can return several valid IP addresses for a host name, the DSLAM software checks the address of the requesting client against all IP addresses for the named host returned by DNS. If the address sent by the requester is invalid because it does not match any address listed with DNS for the host name, then the DSLAM software rejects the remote command execution request.

Note that if no DNS servers are configured for the DSLAM, then the DSLAM cannot authenticate the host in this manner. In this case, the DSLAM software sends a broadcast request to attempt to gain access to DNS services on another server. If DNS services are not available, you must use the **no ip domain-lookup** command to disable the attempt of the DSLAM to gain access to a DNS server by sending a broadcast request.

If DNS services are not available and, therefore, you bypass the DNS security check, the DSLAM software accepts the request to remotely execute a command *only if* all three values sent with the request match exactly the values configured for an entry in the local authentication file.

If DNS is enabled but you do not want to use DNS for rcmd (remote command) queries, use the **no ip rcmd domain-lookup** command.

To ensure security, the DSLAM is *not* enabled to support rcp requests from remote users by default. When the DSLAM is not enabled to support rcp, the authorization database has no effect.

To configure the DSLAM to allow users on remote systems to copy files to and from the DSLAM and execute commands on the DSLAM, perform the tasks in either of the first sections and, optionally, the task in the third section:

- Configure the DSLAM to Accept rcp Requests from Remote Users
- Configure the DSLAM to Allow Remote Users to Execute Commands Using rsh
- Turn Off DNS Lookups for rcp and rsh

Configuring the DSLAM to Accept rcp Requests from Remote Users

To configure the DSLAM to support incoming rcp requests, complete these tasks in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**ip rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username*	Create an entry in the local authentication database for each remote user who is allowed to execute **rcp** commands on the DSLAM.
3.	DSLAM(config)#**ip rcmd rcp-enable**	Enable the DSLAM to support incoming rcp requests.

To disable the DSLAM from supporting incoming rcp requests, use the **no ip rcmd rcp-enable** command.

Note When the DSLAM's support for incoming rcp requests is disabled, you can still use the **rcp** commands to copy images from remote servers. The DSLAM's support for incoming rcp requests is distinct from its ability to handle outgoing rcp requests.

Example

This example shows how to add two entries for remote users to the authentication database of the DSLAM, then enable the DSLAM to support remote copy requests from remote users. Users *netadmin1* on the remote host at IP address 131.108.15.55 and user *netadmin3* is on the remote host at IP address 131.108.101.101. Both are allowed to connect to the DSLAM and remotely execute rcp commands after the DSLAM is enabled to support rcp. Both authentication database entries give the DSLAM's host name *DSLAM1* as the local username. The last command enables the DSLAM to support rcp requests from remote users.

```
DSLAM(config)#ip rcmd remote-host DSLAM1 131.108.15.55 netadmin1
DSLAM(config)#ip rcmd remote-host DSLAM1 131.108.101.101 netadmin3
DSLAM(config)#ip rcmd rcp-enable
```

Configuring the DSLAM to Allow Remote Users to Execute Commands Using rsh

To configure the DSLAM as an rsh server, complete these tasks in global configuration mode:

Step	Command	Task
1.	DSLAM#**configure terminal**	Go to global configuration mode.
2.	DSLAM(config)#**ip rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username* [**enable**]	Create an entry in the local authentication database for each remote user who is allowed to execute rsh commands on the DSLAM.
3.	DSLAM(config)#**ip rcmd rsh-enable**	Enable the DSLAM to support incoming rsh commands.

To disable the DSLAM from supporting incoming **rsh** commands, use the **no ip rcmd rsh-enable** command.

Note When the DSLAM is disabled, you can still issue **rsh** commands to be executed on other DSLAMs that support the rsh protocol and on UNIX hosts on the network.

Example

This example shows how to add two entries for remote users to the authentication database of the DSLAM, and enable the DSLAM to support **rsh** commands from remote users. Users *rmtnetad1* and *netadmin4* are both on the remote host at IP address 131.108.101.101. Although both users are on the same remote host, you must include a unique entry for each user. Both users are allowed to connect to the DSLAM and remotely execute **rsh** commands after the DSLAM is enabled for rsh. User *netadmin4* is allowed to execute privileged EXEC mode commands on the DSLAM. Both authentication database entries give the DSLAM's host name *DSLAM1* as the local username. The last command enables the DSLAM to support **rsh** commands issued by remote users.

```
DSLAM(config)#ip rcmd remote-host DSLAM1 131.108.101.101 rmtnetad1
DSLAM(config)#ip rcmd remote-host DSLAM1 131.108.101.101 netadmin4 enable
DSLAM(config)#ip rcmd rsh-enable
```

Turning Off DNS Lookups for rcp and rsh

To bypass the DNS security check when DNS services are configured but not available, use this command in global configuration mode:

Command	Task
`DSLAM(config)#no ip rcmd domain-lookup`	Bypass the DNS security check.

The DSLAM software accepts the request to remotely execute a command only if all three values sent with the request match exactly the values configured for an entry in the local authentication file.

Configuring the Remote Username for rcp Requests

From the DSLAM, you can use rcp to remotely copy files to and from network servers and hosts if those systems support rcp. You do not need to configure the DSLAM to issue rcp requests from the DSLAM using rcp. However, to prepare to use rcp from the DSLAM for remote copying, you can perform an optional configuration process to specify the remote username to be sent on each rcp request.

The rcp protocol requires that a client send the remote username on an rcp request. By default, the DSLAM software sends the remote username associated with the current TTY (terminal) process, if that name is valid, for **rcp** commands.

If the username for the current TTY process is not valid, the DSLAM software sends the host name as the remote username. For **boot** commands using rcp, the DSLAM software sends the DSLAM host name by default. You cannot explicitly configure the remote username.

If the remote server has a directory structure, as do UNIX systems, rcp performs its copy operations as follows:

- When copying from the remote server, rcp searches for the system image or configuration file to be copied to the directory of the remote username.

- When copying to the remote server, rcp writes the system image or configuration file to be copied to the directory of the remote username.
- When booting an image, rcp searches the directory of the remote username for the image file on the remote server.

To override the default remote username sent on rcp requests, use this command in global configuration mode:

Command	Task
DSLAM(config)#**ip rcmd remote-username** *username*	Specify the remote username.

To remove the remote username and return to the default value, use the **no ip rcmd remote-username** command.

Remotely Executing Commands Using rsh

You can use the **rsh** command to execute commands remotely on network servers that support the remote shell protocol. To use this command, the *.rhosts* files on the network server must include an entry that permits you to remotely execute commands on that host.

If the remote server has a directory structure, as do UNIX systems, the **rsh** command that you issue is remotely executed from the directory of the account for the remote user that you specify through the **/user** *username* keyword and argument pair.

If you do not specify a username, the DSLAM sends a default remote username. By default, the DSLAM software sends the remote username associated with the current TTY process, if that name is valid. If the TTY remote username is invalid, the DSLAM software uses the DSLAM host name as both the remote and local user names.

To execute a command remotely on a network server using rsh, perform these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**enable** [*password*]	Enter privileged EXEC mode.
2.	DSLAM#**rsh** {*ip-address* | *host*} [**/user** *username*] *remote-command*	Enter the command to be executed remotely.

Example

This example shows how to execute a command remotely using rsh:

```
DSLAM>enable
DSLAM#rsh mysys.cisco.com /u sharon ls -a
.
..
.alias
.cshrc
.emacs
.exrc
.history
.login
.mailrc
```

```
.newsrc
.oldnewsrc
.rhosts
.twmrc
.xsession
jazz
DSLAM#
```

Manually Loading a System Image from ROM Monitor

If your DSLAM does not find a valid system image, or if its configuration file is corrupted at startup and the configuration register is set to enter ROM monitor mode, the system might enter ROM monitor mode. From this mode, you can manually load a system image from Flash memory, from a network server file, or from ROM. You can also enter ROM monitor mode by restarting the DSLAM and then pressing the **Break** key during the first 60 seconds of startup.

These sections describe how to manually load a system image from ROM monitor mode:

- Manually Boot from Flash Memory
- Manually Boot from a Network File
- Manually Boot from ROM

Manually Booting from Flash Memory

To manually boot from Flash memory, complete these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**reload**	Restart the DSLAM.
2.	**Break**	Press the **Break** key during the first 60 seconds while the system is starting up.
3.	DSLAM#**boot** *bootflash*:[*filename*]	Manually boot the DSLAM from Flash.

Examples

In this example, the DSLAM is manually booted from Flash memory. Because the optional *filename* argument is absent, the first file in Flash memory is loaded.

```
>boot
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Uncompressing file:  ############################################################
############################################################################
############################################################################
############################################################################
############################################################################
############################################################################
############################################################################
#########

<information deleted>

%SYS-5-RESTART: System restarted --
Cisco Internetwork Operating System Software
```

```
<information deleted>
```

In this example, the **boot bootflash** command is used with the filename 6260-m_1, the name of the file that is loaded:

```
>boot bootflash: 6260-m_1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
Uncompressing file:  ##################################################################
##############################################################################################
##############################################################################################
##############################################################################################
##############################################################################################
##############################################################################################
##############################################################################################
#########

<information deleted>

%SYS-5-RESTART: System restarted --

<information deleted>
```

Manually Booting from a Network File

To manually boot from a network file, complete these tasks in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**reload**	Restart the DSLAM.
2.	**Break**	Press the **Break** key during the first 60 seconds while the system is starting up.
3.	DSLAM#**boot** *filename* [*ip-address*]	Manually boot the DSLAM from a network file.

Note The BOOTLDR variable must be configured to either **bootflash:** *filename* to allow manually booting from a network file. See the "BOOTLDR Environment Variable" section on page 17-39.

Example

In this example, the DSLAM is manually booted from the network file "network1":

```
>boot network1
```

Manually Booting from ROM

To manually boot the DSLAM from ROM, complete these steps in privileged EXEC mode:

Step	Command	Task
1.	DSLAM#**reload**	Restart the DSLAM.

2.	**Break**	Press the **Break** key during the first 60 seconds while the system is starting up.
3.	DSLAM#**boot**	Manually boot the DSLAM from ROM.

Example

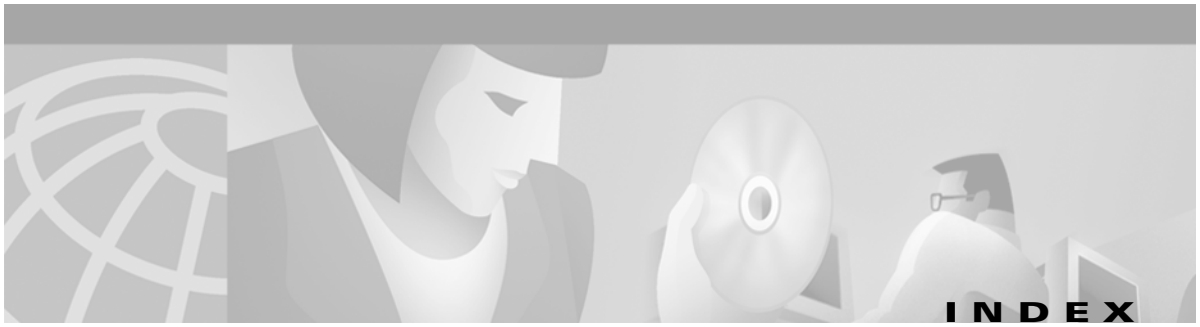
In this example, the DSLAM is manually booted from ROM:

```
>boot
```

Using the System Image Instead of Reloading

To return to EXEC mode from ROM monitor mode, use this command:

Command	Task
DSLAM>**continue**	Return to EXEC mode to use the system image.



INDEX

Symbols

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W