AISwitch™

AI2524 Router

User's Manual

August 1997

Applied Innovation Inc.

AI2524 Router

User’s Manual

AISwitch™

Applied Innovation Inc.

AI2524 Router User's Manual

August 1997
Reference 2524UM

Copyright Notice

FCC Warning

The Federal Communications Commission has set limits for emitted radio interference. The AISwitch is constructed with this electromagnetic interference (EMI) limitation in mind. Th AISwitch is classified under FCC regulations as a Class A device, that is, a device for use in commercial environments and not in residential areas. This device has been tested and shown to comply with the following FCC rule: Part 15 Subpart J. Operation of this equipment in a residential are may cause interference to radio and TV reception, requiring the user to take whatever steps are necessary to correct the interference.

Information is available from the FCC describing possible corrective actions. For lower EMI levels, we suggest using only metal connectors and shielded cables grounded to the frame.

Electrostatic Discharge Warning

Warning: ***The AISwitch and its peripherals contain electrostatic sensitive components. Proper handling, shipping, and storage precautions must be exercised:***

- ***Removal and installation of circuit boards must be performed in a static-free environment. Tthe technician should wear an anti-static wrist strap and stand on an anti-static mat. Both the wrist strap and mat must be grounded at the same point as the AISwitch enclosure.***
- ***When not in use, circuit boards must be kept in their anti-static plastic bags.***
- ***Circuit boards must only be removed from their anti-static plastic bags immediately prior to installation into the AISwitch enclosure.***
- ***Immediately upon removal from the enclosure, circuit boards must be inserted into their anti-static bags.***
- ***Do not ship or store the electronic circuit boards near strong electrostatic, electromagnetic, magnetic, or radioactive fields.***

Specifications are subject to change without notice.

Applied Innovation Inc.
5800 Innovation Driv
Dublin, Ohio 43016-3271
614-798-2000
800-247-9482
FAX: 614-798-1770

Contents

Chapter 1: Introduction

Documentation Overview

This manual documents the use and operation of the AI2524 in an AISwitch system. These topics are covered:

Chapter 1	Introduction This chapter provides a documentation overview, related documentation, contact infor mation, and text conventions.
Chapter 2	AI2524 Overview This chapter describes the Cisco IOS software features and AI2524 hardware specifications.
Chapter 3	Configuration Overview This chapter describes the process of booting the router for the first time and provides overviews of configuration methods.
Chapter 4	Understanding the User Interfac This chapter introduces Cisco IOS softwar interface, and includes instructions for accessing command modes, context-sensitiv help, and command history and editing features.
Chapter 5	Using AutoInstall This chapter describes how use AutoInstall for automatic and dynamic configuration o the AI2524.
Chapter 6	Using the System Configuration Dialog This chapter describes how to use the System Configuration Dialog to manually configure the router.
Chapter 7	Manually Loading System Images This chapter describes how to manually load system images in the event that typical start-up procedures malfunction.

Chapter 8

AI2524 Protocol Configuration Steps

This chapter describes how to configure th AI2524 for OSPF TCP/IP, IGRP TCP/IP, and RIP TCP/IP.

Chapter 9

AI2524 OSI/CLNP Configuration Steps

This chapter describes how to configure th AI2524 for ISO IGRP dynamic routing, IS-IS dynamic routing, and CLNS static routing.

Chapter 10

Serial Interface Configuration Steps

This chapter describes how to configure synchronous serial interfaces.

Chapter 11

AI2524 Sync PPP Configuration Steps

This chapter describes how to enable PPP encapsulation and perform a variety of PPP configuration tasks.

Chapter 12

AI2524 X.25 Configuration Steps

This chapter describes how to configure th X.25 interface, and includes datagram transport, routing, and X.29 access lists and profile scripts.

Chapter 13

AI2524 Frame Relay Configuration Steps

This chapter describes how to enable frame relay encapsulation and perform a variety of Frame Relay configuration tasks.

Chapter 14

T1 Interface Configuration Steps

This chapter describes how to configure th AI2524 for fractional T1.

Chapter 15

56/64-kbps Switched and Digital Data Services (DDS) Interface Configuration Steps

This chapter describes how to configure switched and DDS interfaces.

Chapter 16

Basic Configuration

This chapter describes the AI2524’s panel, cable connections, and firmware.

Chapter 17

Command References

This chapter contains a link to the AI2524/ Cisco IOS v. 11.2 documentation CD.

Chapter 18 System Error Messages

This chapter contains a link to the AI2524/ Cisco IOS v. 11.2 documentation CD.

Chapter 19 Debug Command Reference

This chapter contains a link to the AI2524/ Cisco IOS v. 11.2 documentation CD.

Appendix A AISwitch Release Note

Includes the release notes for this version.

Appendix B Acronyms

This appendix defines acronyms used in this manual.

Related Documentation

Documentation for AISwitch products includes:

AI120 Contact Alarm Monitor User's Manual	AI120UM
AISwitch 130 Hardware Manual	HM130-0194
AI192-X User's Manual	192UM
AI193-ES User's Manual	UM193ES
AI193-TX User's Manual	193TXUM
AI194 User's Manual	AI94U
AI196-I User's Manual	196IUM
AI196-IEGB User's Manual	196TUM
AI196-X User's Manual	196XUM
AI198 System Manager/User's Manual	98UM
Common Alarm Panel Manual	180CAPUM
RDC180HP Power Supply Manual	180PSUM
AI325AC Power Supply Manual	AI325UM
AI180FRF AICool Fan Manual	HMFRF-A
AISwitch 180 Hardware Manual	HW0593
AppliedView Network Management System User's Manual	AV201UM

To order these or any other AISwitch manuals, contact your sales representative at (800) 247-9482.

Contact Information

To register documentation, contact Applied Innovation Inc. at:

Applied Innovation Inc.
Publications Dept.
5800 Innovation Dr.
Dublin, OH 43216-3271

Phone	(614) 798-2000 (800) 247-9482
FAX	(614) 798-1770
Email	aidoc@aiinet.com

Register your documentation by completing the registration form.

The most current version of release notes and the *AI198 CLC User’s Manual* are available on the Applied Innovation web site at:

http://www.aiinet.com

Click on the Documentation option. These additional email contacts are also available:

National Sales Department	sales@aiinet.com
Customer Service Department	cssupport@aiinet.com
Marketing Department	info@aiinet.com
Human Resources Department	hr@aiinet.com
Technical Support Department	techsupp@aiinet.com
Investment Relations	invest@aiinet.com
Feedback for Engineering (R&D)	feedback@aiinet.com
About SNMP	snmp@aiinet.com
Webmaster	webmaster@aiinet.com
Documentation Department	aidoc@aiinet.com

Text Conventions

Important concepts throughout this manual are emphasized with these special text styles:

[Buttons]	Function buttons that appear on a screen ar shown in regular body text and enclosed in square brackets. For example: [Close] [Send]

`Commands`	In command lines, type text that appears in this style exactly as shown:
	`avdumpdb`
	`BNC OFF`
	Press the <Return> or <Enter> key after all commands.

Variable Arguments	Variable arguments are text that you specify. They are shown in italics. For example: **`avaccess`** ***`switch_name`*** In this case, “switch_name” is variable text. To enter the command, type **`avaccess`** and then the actual name of the switch.
...	Ellipses (...) signify that the preceding argument can be repeated a number of times. For example: **`cat`** ***`filename...`*** means that you would typ **`cat`** followed by one or more filenames.

`[Optional Arguments]`	Some arguments are optional. This means that you have the choice of including them or not. Optional arguments are shown enclosed in square brackets, which are not entered. For example: **avrestore [*directory*]** means that you type **avrestore** and (if you need to include a directory) type the actual path name. **CFGMSG *n*, [DEFAULT]** means that you type **CFGMSG** followed by message number, a comma, and (optionally) the word **DEFAULT** without brackets.
`{argument| argument}`	Arguments between braces are grouped into one unit. The vertical bar signifies that either the first or second argument can be used. The braces and vertical bar are not entered. For example: **ls {*file* | *directory*}** means that you would type **ls** followed by either a file or a directory name.
<Keys>	Keyboard controls are shown in this style. Angle brackets depict keys that do not appear on the screen when pressed, such as the <tab> or <return> keys. Keys used in combination are connected with a dash. For example, to enter: <ALT-SysRq> hold down the Alt key while you press th SysRq key.

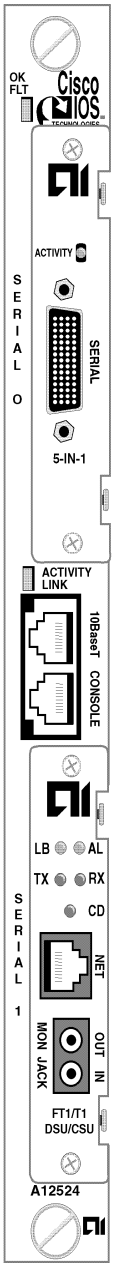
Labels	Labels are used in diagrams to designat physical components such as jumper straps, switches, and cable connectors. For example: **COM1** **BOOT1** If the physical component being described is part of the text, it appears as regular type: To reset the COM1 port connector, press th BOOT switch.
Menu | Submenu	Menu selections are shown in bold text. The bar separates the main menu from submenus. For example: **File | Exit** indicates that you should select the File menu, and then select the Exit menu item.

`Screen output`	Screen shots, system prompts, and error messages displayed on the screen are shown in this style:
	`+CONFIG PORT,LPORT=40,HPORT=47,BITS=8`
	`+CONFIG PORT,LPORT=48,HPORT=49,BITS=7`
	`+CONFIG PORT,LPORT=50,BITS=8$0778`

Warning:	Warning messages indicate critical information required for your safety or for correct system operation. For example: ***Warning: Failure to heed this important text could cause damage or unreliable results.***

Chapter 2: AI2524 Overview

Introduction



This chapter describes the Cisco IOS software features and the AI2524 hardware specifications. The AI2524 multi-protocol router adds the Cisco Internetworking Operating System (IOS™) routing software to the NEBS compliant AISwitch 180 Series. The AI2524 adds leading edge routing capabilities through TCP/IP and OSI networks to the AI 180 Switch protocol conversion and data port concentration capabilities.

The AI2524 is designed for reliable connectivity to a Wide-Area Network (WAN). It can provide WAN/DCN connections to centralized operations support systems from central office based LAN/LCN or X.25 networks.

The AI2524 has one Ethernet LAN interface with both a backplane (IRB) port and a 10BaseT port on the front panel. The backplane port provides connections to a large number of interfaces and protocols through a variety of AISwitch interface modules.

The base module accepts two serial interface modules in any combination. The three available modules are:

T1 CSU/DSU

The AI2524-T1 module is an integrated CSU/DSU that supports full or fractional T1 leased line services. It may be remotely managed using Simple Network Management Protocol (SNMP)

Four Wires 56K CSU/DSU

The AI2524-4W56 module is an integrated CSU/DSU that supports 4-wire 56k leased line or switched services. It may be remotely managed using Simple Network Management Protocol (SNMP)

5-IN-1 Serial Cable Interface

The AI2524-5N1 module provides a cable interface to a synchronous serial line. It supports full and half duplex operations up to 2.048 MHz, full duplex. Dependent upon the interface, DTE/DCE and NRZ/NRZ1 operations are available. Cables are available for the these interfaces:

RS-232 DTE or DCE, EIA-530 DTE, RS-449 DTE or DCE, OR V.35, DTE or DCE, up to E1 speeds.

Software Features and Functions

The AI 2524 incorporates Cisco IOS software. This software provides:

- Scalability
- Reliable, Adaptive Routing
- Remote Access and Protocol Translation
- Management and Security

Scalability

The Cisco IOS software uses scalable routing protocols to avoid needless congestion, overcome inherent protocol limitations, and bypass many of the obstacles that result from the complex scope and geographical dispersion of an internetwork.

The Cisco IOS software eliminates the need for static routes and reduces network costs by efficiently using network bandwidth and resources. Advanced features such as route filtering, protocol termination and translation, smart broadcasts, and helper address services combine to create a flexible, scalable infrastructure that can keep pace with evolving network requirements.

Reliable, Adaptive Routing

The AI2524 Cisco IOS software identifies the best network paths and routes traffic around network failures. Policy-based features such as route filtering and route redistribution save network resources by preventing data from being broadcast to nodes that do not need it. Priority output queuing and custom queuing grant priority to important sessions when network bandwidth is scarce. Load balancing uses every available path across the internetwork to preserve bandwidth and improve network performance. The Cisco IOS software also provides the most effective and efficient scaling available for network applications that require transparent or source-route bridging algorithms.

Remote Access and Protocol Translation

Your router connects terminals, modems, microcomputers, and networks over serial lines to LANs or Wide-Area Networks (WANs). It also provides network access to terminals, printers, workstations, and other networks.

On LANs, terminal services support Transmission Control Protocol/ Internet Protocol (TCP/IP) on UNIX machines with Telnet and rlogin connections. You can use the router to make connections between hosts and resources running different protocols, including router and access server connections to X.25 machines using X.25 Packet Assembler/Disassembler (PAD).

The Cisco IOS software supports three types of server operation:

Remote Node Services	Connect devices over a telephone network Serial Line Internet Protocol (SLIP), compressed SLIP (CSLIP), Point-to-Point Protocol (PPP), and X-Windows terminal protocol. See Figure 2- .
Terminal Service	Connect asynchronous devices to a LAN or WAN through network and terminal-emulation software including Telnet and rlogin,.
Protocol Translation Services	Convert one virtual terminal protocol into another protocol. See Figure 2-1.

Figure 2-1:Remote Access Functionality

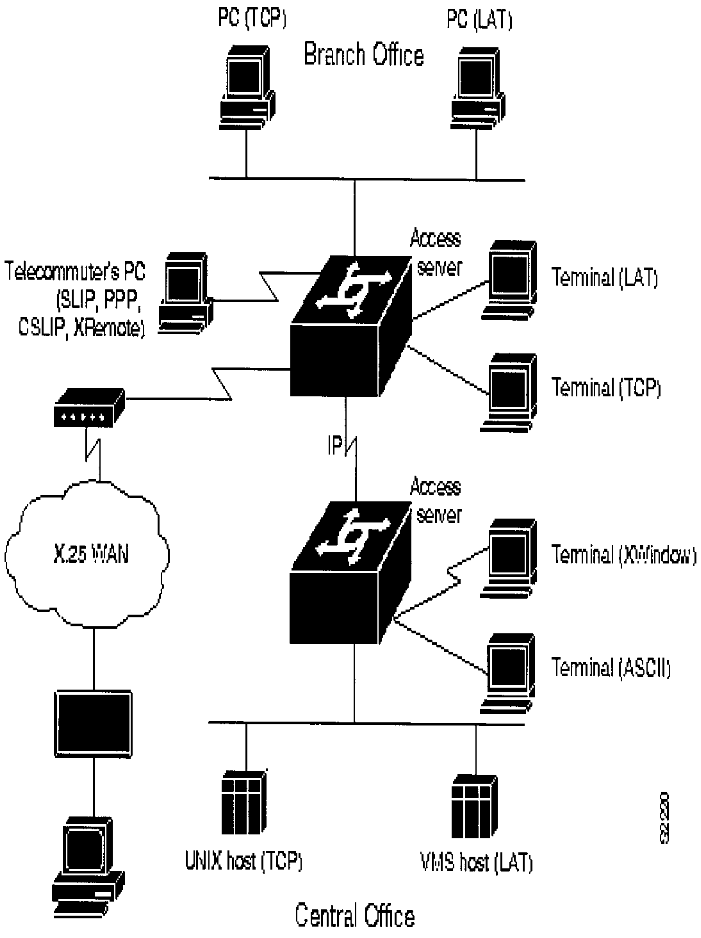


Figure 2-1 illustrates the functions available on access servers:

- Remote node service is demonstrated by the telecommuter’s (remote) PC connection running SLIP, PPP, CSLIP, or XRemote
- Terminal service is shown between the terminals and hosts run ning the same protocol (LAT-to-LAT or TCP-to-TCP)
- Protocol translation is shown between the terminals and hosts running unlike protocols (LAT-to-TCP or TCP-to-LAT)
- Asynchronous IP routing is shown by the PC running SLIP or PPP, and between the two access servers.

Management and Security

The Cisco IOS software provides an array of network management and security capabilities. Integrated management simplifies administrative procedures and shortens the time required to diagnose and fix problems. Automated operations reduce hands-on tasks and make it possible to manage large, geographically dispersed internetworks with a small staff of experts located at a central site.

The Cisco IOS software provides several management features including configuration services, which lower the cost of installing, upgrading, reconfiguring routers, and reconfiguring access servers, as well as comprehensive monitoring and diagnostic services. In addition, the Cisco IOS software provides information and services for router management applications.

Management services are matched by their security capabilities. Th Cisco IOS software includes a diverse tool kit for partitioning resources and prohibiting access to sensitive or confidential information and processes. Multidimensional filters prevent users from knowing that other users or resources are on the network. Encrypted passwords, dial-in authentication, multilevel configuration permissions, network data encryption, and accounting and logging features provide protection from and information about unauthorized access attempts and data eavesdropping attempts.

Software Specifications

Supported Media

The AI2524 supports these industry-standard networking media:

- Channelized T1
- Ethernet: IEEE 802.3 and Type I
- Fiber Distributed Data Interface (FDDI): single and dual mode
- High-Speed Serial Interface (HSSI): supports T1
- Synchronous serial: V.35, RS-232, RS-449, and RS-530

Supported Network Protocols

The Cisco IOS software supports many networking protocols, as well as their associated routing protocols. These protocols are based on both open standards and proprietary protocols from a variety of vendors.

The Cisco IOS software can receive and forward packets concurrently from any of these combinations:

WAN protocols

- Frame Relay
- High-Level Data Link Control (HDLC)
- PPP
- X.25 a

Network protocols

- IP
- OSI Connectionless Network Services (CLNS) and Connection Mode Network Services (CMNS)

IP Routing Protocols

The Cisco IOS software supports IP routing protocols, including interior gateway protocols and exterior gateway protocols.

Interior Gateway Protocols

- Internet Gateway Routing Protocol (IGRP)
- Enhanced IGRP
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP) and RIP Version 2
- Intermediate System-to-Intermediate System (IS-IS

Exterior Gateway Protocols

- Exterior Gateway Protocol (EGP)
- Router Discovery Protocols
- ICMP Router Discovery Protocol (IRDP)
- Hot Standby Router Protocol (HSRP)

Connections

External Connection Requirements

The AI2524 provides LAN and WAN access in a modular router platform. The router includes an Ethernet (AUI or 10BaseT) LAN connection, and accommodates two synchronous serial modules.

The synchronous serial WAN modules include these external connectors:

- Four-wire 56/64-kbps DSU/CSU WAN module with an RJ-48S connector
- Fractional T1/T1 DSU/CSU WAN module with an RJ-48C connector
- Five-in-one synchronous serial WAN module with a DB-60 serial connector. The five-in-one synchronous serial interface supports the following signaling standards: EIA/TIA-232, EIA/TIA-449, V.35, X.21, and EIA-530

Chapter 3: Configuration Overview

Introduction

This chapter provides a brief overview of three ways that the AI2524 can be configured including:

- Configuration Mode
- AutoInstall
- System Configuration Dialog

These three and other procedures are described in detail in this manual.

The first time the router is powered and booted, you must enter basic configuration information and save the configuration to a file in NVRAM.

Boot Router for First Time

Each time you power on the router, it goes through the boot sequence:

1. The router goes through power-on self-test diagnostics to verify basic operation of the CPU, memory, and interfaces.
2. The system bootstrap software (boot image) executes and searches for a valid Cisco IOS image (router operating system software). The source of the Cisco IOS image (Flash memory or a Trivial File Transfer Protocol [TFTP] server) is determined by the configuration register setting. The factory-default setting for the configura tion register is 0x2102, which indicates that the router should attempt to load a Cisco IOS image from Flash memory.
3. If after five attempts a valid Cisco IOS image is not found in Flash memory, the router reverts to boot ROM mode (which is used to install or upgrade a Cisco IOS image).
4. If a valid Cisco IOS image is found, then the router searches for a valid configuration file.

5. If a valid configuration file is not found in NVRAM, the router runs the System Configuration Dialog so you can configure it manually. For normal router operation, there must be a valid Cisco IOS image in Flash memory and a configuration file in NVRAM.

   The first time you boot your router, you will need to configure the router interfaces and then save the configuration to a file in NVRAM.

Configure the Router

You can configure the router using one of these procedures:

- Configuration mode: Recommended if you are familiar with Cisco IOS software commands. Refer to the Understanding the User Interface chapter.
- AutoInstall: Recommended for automatic installation if another router running Cisco IOS software is installed on the network. This configuration method must be set up by someone with experience using Cisco IOS software. Refer to the Use AutoInstall chapter.
- System Configuration Dialog: Recommended if you are not familiar with Cisco IOS commands. Refer to th Using the System Configuration Dialog chapter.

Use the procedure that best fits the needs of your network configura tion and level of Cisco IOS software experience.

Using Configuration Mode

You can configure the router manually if you prefer not to use AutoInstall or the System Configuration Dialog. Take these steps to configure the router manually:

1. Connect a console terminal to the AI2524 and power ON th router.

2. When you are prompted to enter the initial dialog, type **no** to go into the normal operating mode of the router:

```
Would you like to enter the initial dialog? [yes]: no
```

3. After a few seconds you will see the user EXEC prompt **(Router>** . Enter the enable command to enter enable mode. You can only make configuration changes in enable mode.

```
Router> enable
```

The prompt changes to the privileged EXEC (enable) prompt:

```
Router#
```

4. Enter the **`configure terminal`** command at the enable prompt to enter configuration mode:

```
Router# configure terminal
```

You can now enter any changes you want to the configuration. Refer to the appropriate sections of this manual for help with specific configurations.

5. Press <Ctrl-Z> to exit configuration mode.

Show Configuration

To see the current operating configuration, enter the **`show running-config`** command at the enable prompt:

```
Router# show running-config
```

To see the configuration in NVRAM, enter th **`show startup-config`** command at the enable prompt:

```
Router# show startup-config
```

The results of the **`show running-config`** and **`show startup-config`** commands will be different if you have made changes to the configuration but have not yet written them to NVRAM.

Save the Configuration

To make your changes permanent, enter the **`copy running-config startup-config`** command at the enable prompt:

```
Router# copy running-config startup-config
```

The router is now configured and will boot with the configuration you entered.

Configuration Overviews

This section describe the software tools you can use to configure your router via the Cisco IOS software:

- Configuration Builder
- Command Interpreter
- Web Browser Interface
- Use ClickStart

Use Configuration Builder

The Configuration Builder allows you to create configuration files for multiple routers or access servers without knowing the command-line language or syntax. It is a Microsoft Windows-based application that runs on an IBM PC or compatible computer.

If you do not have the platform required to run Configuration Builder, configure your device using the command interpreter.

Use the Command Interpreter

You can build most straightforward configurations and create a configuration file using the setup command facility. Refer to the Using AutoInstall chapter for more information.

Before configuring your router or access server, you must determine these items:

- Which network protocols you are supporting (for example, IP and Novell IPX)
- The addressing plan for each network protocol

- Which WAN protocols you will run on each interface (for exam ple: Frame Relay, HDLC, SMDS, and X.25)
- Which routing protocol you will use for each network protocol

The Cisco IOS software provides a user interface called a command interpreter, or EXEC, that allows you to configure and manage th router or access server. The user interface also provides context-sensitive help. The command interpreter has several command modes, each of which provides a group of related commands that you can use to configure the routing device and display its status. Some commands are available to all users, others can be executed only after the user enters an enabling password. Context-sensitive help gives information about command syntax. The command interpreter and its help feature are described in the Understanding the User Interface chapter.

You use the command interpreter (also known as the command-line parser) to configure interfaces, terminal sessions, and asynchronous communications lines. Interfaces are connections to network media, such as Ethernet, Token Ring, and serial media. You configure them to run routing and networking protocols. You configure terminal sessions and modems connected to the router or access server so that other network users can log in to the network over asynchronous lines.

You can configure and manage the router or access server, performing such tasks as naming the device, setting the time, configuring SNMP, and setting security.

Follow this basic process to set up your access server or router:

1. Attach an RS-232 ASCII terminal to the system console port lo cated at the rear of the router.
2. Configure the terminal to operate at 9600 baud, 8 data bits, no parity, 2 stop bits.
3. Power up the router. The setup command facility runs automati cally for initial startup.
4. Perform general system configuration.
5. Configure your system by referring to the appropriate part in th documentation.

To enhance the configuration, perform the protocol-specific tasks described in the appropriate chapters of this guide.

Use the Web Browser Interface

You can issue most of the AI2524 commands using a Web browser. You access the Web Browser interface through the router's home page. AI2524 routers loaded with the latest version of the Cisco IOS software have a home page, which is password protected.

From the router's home page, click on the Monitor the Router hyper text link. This link takes you to a Web page that has a Command field. You can type commands in this field as if you were using the com mand interpreter on a terminal connected to the router. The page also displays a list of hypertext commands that can be executed with a mouse-click. This feature is documented in Web Browser Interface.

Configuration Storage and Hot Swap

The AI2524 card must have its configuration stored on the AI198 card to insure the ability to perform hot swap. The configuration is stored in Menu 4.18 for the port associated with the AI2524 card. When the AI2524 card boots, it uses the BOOTP protocol to obtain its IP address. The AI2524 card then uses TFTP to transfer its configuration information from the AI198 card to the AI2524 card. Any changes made to the AI2524 card's configuration must be stored back to the AI198 card to maintain full hot swap capability. Here are two options for managing the AI2524 card configuration:

Always Modify the Configuration Using Menu 4.18

This option suggests that you use Menu 4.18 whenever you mak modifications to the AI2524 card configuration. After you complete the modifications, you can use the **`ENABL`** command to reset the AI2524 card. This allows the changes to take affect.

Store the Configuration on the AI198 Card

This option suggests that whenever you make modifications to the AI2524 card configuration using the AI2524 configuration mecha nism (telnet to the card and change the configuration, Quick Start, CiscoWorks), you must store the modifications on the AI198 card. You can save the configuration to the AI198 card using TFTP on the AI2524 card.

Chapter 4: Understanding the User Interface

Introduction

This chapter describes the features of the user interface.The AI2524 user interface provides access to several different command modes. Each command mode provides a group of related commands. This chapter describes how to access and list the commands available in each command mode and explains the primary uses for each command mode.

Command Line Interface

AI2524 commands can be entered at a terminal connected to the router using the command line interface (CLI). Commands may also be entered using the Web Browser interface. All routers using the Cisco IOS software have a home page. From this home page, you can access the Web Browser interface, which allows you to execute AI2524 commands. You can execute these commands by clicking on them or entering them in a command field. This feature is described in Web Browser Interface.

For security purposes, the Cisco IOS software provides two levels of access to commands: user and privileged. The unprivileged user mod is called user EXEC mode. The privileged mode is called privileged EXEC mode and requires a password. The commands available in user EXEC mode are a subset of the commands available in privileged EXEC mode.

If your router or access server does not find a valid system image, or if its configuration file is corrupted at startup, the system might enter read-only-memory (ROM) monitor mode. ROM monitor mode can also be accessed through privileged EXEC mode.

From privileged EXEC mode, you can access the global configuration mode and a number of specific configuration modes. These modes are listed in Command Modes.

Entering a question mark (?) at the system prompt allows you to obtain a list of commands available for each command mode.

Almost every configuration command also has a **no** form. In general, use the **no** form to disable a feature or function. Use the command without the keyword **no** to reenable a disabled feature or to enable a feature that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, specify the **`no ip routing`** command and specify **`ip routing`** to reenable it.

The user interface also provides context-sensitive help for command syntax. This chapter describes how to use the help system. It also de scribes the command editing and command history features that enable you to recall and easily edit command entries.

End a Session

After using the **`setup`** command or other configuration commands, exit the user interface and quit the session. To end a session, type:

```
quit
```

User Interface Task List

You can perform these tasks to familiarize yourself with the AI2524 user interface:

- Access Each Command Mode
- Get Context-Sensitive Help
- Check Command Syntax
- Use the Command History Features
- Use the Editing Features
- End a Session
- Use the Web Browser Interface to Issue Commands

Command Modes

This section describes how to access each of the AI2524 command modes, including:

- User EXEC Mode Commands
- Privileged EXEC Mode Commands
- ROM Monitor Mode Commands
- Global Configuration Mode Commands

These command modes are accessible from global configuration mode:

- Interface Configuration Mode Commands
- Subinterface Configuration Mode Commands
- Router Configuration Mode
- IPX-Router Configuration Mode
- Route-Map Configuration Mode
- Key Chain Configuration Mode
- Key Chain Key Configuration Mode
- Response Time Reporter Configuration Mod
- Access-List Configuration Mode

The following table lists the command modes, how to access each mode, the prompt while in each mode, and how to exit each mode. The prompts listed assume that the default device name is Router unless it has been changed during initial configuration using the **`setup`** command. Refer to the product user guide for information on the setup facility. You can also change the host name using the **`hostname`** global configuration command. The following table does not include all o the possible ways to access or exit each command mode.

For all command modes, typing a question mark (?) at the prompt will list the commands available. Refer to Context-Sensitive Help, for more information.

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	**Router>**	Use the **logout** command.
Privileged EXEC	From user EXEC mode, type **enable** (requires a password).	**Router#**	To exit back to user EXEC mode, type **disable**. To enter global configuration mode, use the **configure** privileged EXEC command.
ROM monitor	From privileged EXEC mode, type **reload**. Press <Break> during the first 60 seconds while the system is booting.	>	To exit to user EXEC mode, type **continue**.
Global configuration	From privileged EXEC mode, type **configure**.	**Router(config)#**	To exit to privileged EXEC mode, type **exit** or **end** or press <Ctrl-Z>. To enter interface configuration mode, enter an interface configuration command.
Interface configuration	From global configuration mode, type **interface** ***type numbe*** .	**Router(config-if)#**	To exit to global configuration mode, type **exit**. To exit to privileged EXEC mode, type **exit** or press <Ctrl-Z>. To enter subinterface configuration mode, specify a subinterface with the **interface** command.
Subinterface configuration	From global configuration mode, type **interface** ***type number***; **encapsulation frame-relay**; **interface** ***type number.subinterface-numbe*** .	**Router(config-subif)#**	To exit to global configuration mode, type **exit**. To enter privileged EXEC mode, type **end** or press <Ctrl-Z>.
Router configuration	From global configuration mode, type **router** ***keyword***.	**Router(config-router)#**	To exit to global configuration mode, type **exit**. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.
IPX-router configuration	From global configuration mode, type **ipx routing** ***keyword***.	**Router(config-ipx-router)#**	To exit to global configuration mode, type **exit**. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.
Route-map configuration	From global configuration mode, type **route-map** ***tag***.	**Router(config-route-map)#**	To exit to global configuration mode, type **exit**. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.
Key chain configuration	From global configuration mode type **keychain** ***name***.	**Router(config-keychain)#**	To exit to global configuration mode, type **exit** command. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.

Command Mode (Contd.)	Access Method	Prompt	Exit Method
Key chain key configuration	From key chain configuration mode, type **key** ***number***.	**Router(config-keychain-key)#**	To exit to key chain configuration mode, type **exit**. To exit to global configuration mode, type **exit** command. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.
Response time reporter configuration	From global configuration mode, type **rtr probe**.	**Router(config-rtr)#**	To exit to global configuration mode, type **exit**. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.
Access-list configuration	From global configuration mode, type **ip access-list** ***mode name***.	**Router(config-std-nacl)#** **or** **Router(config-ext-nacl)#**	To exit to global configuration mode, type **exit**. To exit to privileged EXEC mode, type **end** or press <Ctrl-Z>.

User EXEC Mode Commands

After you log in to the router or access server, you are automatically in user EXEC command mode. The EXEC commands available at th user level are a subset of those available at the privileged level. In general, the user EXEC commands allow you to connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and list system information.

```
Router> ?
Exec commands:
<1-99>                  Session number to resume
connect                 Open a terminal connection
disconnect              Disconnect an existing telnet session
enable                  Turn on privileged commands
exit                    Exit from the EXEC
help                    Description of the interactive help system
lat                     Open a lat connection
lock                    Lock the terminal
login                   Log in as a particular use
logout                  Exit from the EXEC
menu                    Start a menu-based user interfac
mbranch                 Trace multicast route for branch of tree
mrbranch                Trace reverse multicast route to branch of tree
mtrace                  Trace multicast route to group
name-connection         Name an existing telnet connection
pad                     Open a X.29 PAD connection
ping                    Send echo messages
resume                  Resume an active telnet connection
show                    Show running system information
systat                  Display information about terminal lines
telnet                  Open a telnet connection
terminal                Set terminal line parameters
tn3270                  Open a tn3270 connection
trace                   Trace route to destination
where                   List active telnet connections
x3                      Set X.3 parameters on PAD
xremote                 Enter XRemote mode
```

Privileged EXEC Mode Commands

Because many of the privileged commands set operating parameters, privileged access should be password protected to prevent unauthorized use.

If the system administrator has set a password, you are prompted to enter it before being allowed access to privileged EXEC mode. Th password is not displayed on the screen and is case sensitive. If a password has not been set, privilege EXEC mode can be accessed only from the router console. The system administrator uses the **enable password** global configuration command to set the password that restricts access to privileged EXEC mode.

The privileged command set includes those commands contained in user EXEC mode, as well as the **configure** command through which you can access the remaining command modes. Privileged EXEC mode also includes high-level testing commands, such as **debug**.

```
Router> enable
Password:
Router# ?
Exec commands:
bfe                      For manual emergency modes setting
clear                    Reset functions
clock                    Manage the system clock
configure                Enter configuration mode
connect                  Open a terminal connection
copy                     Copy a config file to or from a tftp server
debug                    Debugging functions
disable                  Turn off privileged commands
disconnect               Disconnect an existing telnet session
enable                   Turn on privileged commands
exit                     Exit from the EXEC
help                     Description of the interactive help system
lat                      Open a lat connection
llc2                     Execute llc2 tests
lock                     Lock the terminal
login                    Log in as a particular use
logout                   Exit from the EXEC
menu                     Start a menu-based user interf
name-connection          Name an existing telnet connection
ping                     Send echo messages
reload                   Halt and perform a cold restart
resume                   Resume an active telnet connection
send                     Send a message to other tty lines
setup                    Run the SETUP command facility
show                     Show running system information
systat                   Display information about terminal lines
telnet                   Open a telnet connection
terminal                 Set terminal line parameters
test                     Test subsystems, memory, and interfaces
tn3270                   Open a tn3270 connection
trace                    Trace route to destination
where                    List active telnet connections
which-route              Do route table lookup and display results
write                    Write running configuration to memory, network, or
                            terminal
x3                       Set X.3 parameters on PAD
xremote                  Enter XRemote mode
```

ROM Monitor Mode Commands

If your router or access server does not find a valid system image, or if you interrupt the boot sequence, the system might enter ROM monitor mode. From ROM monitor mode, you can boot the device or perform diagnostic tests.

You can also enter ROM monitor mode by entering the **reload** EXEC command and then pressing <Break> during the first 60 seconds of system startup. To save changes to the configuration file, us the **copy running-config startup-config** command before issuing the **reload** command.

```
> ?
$ state              Toggle cache state (? for help)
B [filename] [TFTP Server IP address | TFTP Server Name]
Load and execute system image from ROM or from TFTP server
C [address]          Continue execution [optional address]
D /S M L V           Deposit value V of size S into location L with modifier
E /S M L             Examine location L with size S with modifier M
G [address]          Begin execution
H                    Help for commands
I                    Initialize
K                    Stack trace
L [filename] [TFTP Server IP address | TFTP Server Name]
Load system image from ROM or from TFTP server, but do not begin execution
O                    Show configuration register option settings
P                    Set the break point
S                    Single step next instruction
T function           Test device (? for help)
Deposit and Examine sizes may be B (byte), L (long) or S (short).
Modifiers may be R (register) or S (byte swap).
Register names are: D0-D7, A0-A6, SS, US, SR, and PC
```

To return to user EXEC mode, type **continue**. To initialize the router or access server, enter th **i** command. Th **i** command causes the bootstrap program to reinitialize the hardware, clear the contents of memory, and boot the system. (Use the **i** command before you run any tests or boot the software.) To boot the system image file, use the **b** command.

Global Configuration Mode Commands

Global configuration commands apply to features that affect the system as a whole. Use the **`configure`** privileged EXEC command to enter global configuration mode. When you enter this command, the system EXEC prompts you for the source of the configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

You can specify either the terminal, nonvolatile random access memory (NVRAM), or a file stored on a network server as the source of configuration commands. The default is to enter commands from th terminal console. Pressing <Enter> begins this configuration method.

```
Router# configure
Configuring from terminal, memory, or network [terminal]? Enter configuration
commands, one per line. End with CNTL/ Z
Router(config)# ?
Configure commands:
access-list                 Add an access list entry
apollo                      Apollo global configuration commands
appletalk                   Appletalk global configuration commands
arp                         Set a static ARP entry
async-bootp                 Modify system bootp parameters
autonomous-system           Specify local AS number to which we belong
banner                      Define a login banner
boot                        Modify system boot parameters
bridge                      Transparent bridging
buffers                     Adjust system buffer pool parameters
busy-message                Display message when connection to host fails
chat-script                 Define a modem chat script
clns                        Global CLNS configuration subcommands
clock                       Configure time-of-day clock
decnet                      Global DECnet configuration subcommands
default-value               Default character-bits values
dialer-list                 Create a dialer list entry
enable                      Modify enable password parameters
end                         Exit from configure mode
exit                        Exit from configure mode
frame-relay                 Global frame relay configuration commands
help                        Description of the interactive help system
hostname                    Set system's network name
interface                   Select an interface to configure
ip                          Global IP configuration subcommands
ipx                         Novell/IPX global configuration commands
line                        Configure a terminal lin
lnm                         IBM Lan Manager
locaddr-priority-list       Establish queueing priorities based on LU address
logging                     Modify message logging facilities
login-string                Define a host-specific login string
mop                         The DEC MOP Server
netbios                     NETBIOS access control filtering
no                          Negate a command or set its defaults
ntp                         Configure NTP
priority-list               Build a priority lis
queue-list                  Build a custom queue lis
rif                         Source-route RIF cache
route-map                   Create route-map or enter route-map command mode
router                      Enable a routing process
scheduler-interval          Maximum interval before running lowest priority process
service                     Modify use of network based services
smt-queue-threshold         Set the max number of unprocessed SMT frames
snmp-server                 Modify SNMP parameters
source-bridge               Source-route bridging ring groups
stun                        STUN global configuration commands
tacacs-server               Modify TACACS query parameters
tftp-server                 Provide TFTP service for netload requests
```

```
tn3270                      tn3270 configuration command
username                    Establish User Name Authentication
vines                       Vines global configuration commands
X.25                        X.25 Level 3
xns                         XNS global configuration commands
```

Interface Configuration Mode Commands

Many features are enabled on a per-interface basis. Interface configuration commands modify the operation of an interface such as an Ethernet, Fiber Distributed Data Interface (FDDI), or serial port. Interface configuration commands always follow an **`interface`** global configuration command, which defines the interface type.

This example shows how to access interface configuration commands for serial interface 0 and how to list the available commands.

```
Router(config)# interface serial 0
Router(config-if)# ?
Interface configuration commands:
access-expression       Build a bridge boolean access expression
apollo                  Apollo interface subcommands
appletalk               Appletalk interface subcommands
arp                     Set arp type (arpa, probe, snap) or timeout
backup                  Modify dial-backup parameters
bandwidth               Set bandwidth informational parameter
bridge-group            Transparent bridging interface parameters
clns                    CLNS interface subcommands
clockrate               Configure serial interface clock speed
custom-queue-list       Assign a custom queue list to an interface
decnet                  Interface DECnet config commands
delay                   Specify interface throughput delay
description             Interface specific description
dialer                  Dial-on-demand routing (DDR) commands
dialer-group            Assign interface to dialer-list
down-when-looped        Force looped serial interface down
encapsulation           Set encapsulation type for an interface
ethernet-transit-oui    Token-ring to Ethernet OUI handling
exit                    Exit from interface configuration mode
frame-relay             Set frame relay parameters
hdh                     Set HDH mode
help                    Description of the interactive help system
hold-queue              Set hold queue depth
ip                      Interface Internet Protocol config commands
ipx                     Novell interface subcommands
isis                    IS-IS commands
iso-igrp                ISO-IGRP interface subcommands
keepalive               Enable keepalive
lapb                    X.25 Level 2 parameters (Link Access Procedure, Balanced)
llc2                    LLC2 Interface Subcommands
lnm                     IBM Lan Manager
locaddr-priority        Assign a priority group
loopback                Configure internal loopback on an interface
mac-address             Manually set interface MAC address
mop                     DEC MOP server commands
mtu                     Set the interface Maximum Transmission Unit (MTU)
netbios                 Use a defined NETBIOS access list or enable name-caching
no                      Negate a command or set its defaults
ntp                     Configure NTP
ppp                     Point-to-point protocol
priority-group          Assign a priority group to an interface
pulse-time              Enables pulsing of DTR during resets
pup                     PUP interface subcommands
sdlc                    SDLC commands
sdllc                   Configure SDLC to LLC2 translation
shutdown                Shutdown the selected interface
smds                    Modify SMDS parameters
source-bridge           Configure interface for source-route bridging
stun                    STUN interface subcommands
transmit-interface      Assign a transmit interface to a receive-only interface
```

```
transmitter-delay        Set dead-time after transmitting a datagram
tunnel                   protocol-over-protocol tunneling
tx-queue-limit           Configure card level transmit queue limit
vines                    Vines interface subcommands
xns                      XNS interface subcommands
```

Subinterface Configuration Mode Commands

You can configure multiple virtual interfaces (called subinterfaces) on a single physical interface. This feature is supported on serial inter faces with Frame Relay encapsulation.

Subinterfaces appear to be distinct physical interfaces to the various protocols. For example, Frame Relay networks provide multiple point-to-point links called permanent virtual circuits (PVCs). PVCs can be grouped under separate subinterfaces that in turn are configured on a single physical interface. From a bridging spanning-tree viewpoint, each subinterface is a separate bridge port, and a frame arriving on one subinterface can be sent out on another subinterface.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, a router or access server can receive an ARPA-framed internet packet exchange (IPX) packet and forward the packet back out the same physical interface as a Subnetwork Access Protocol (SNAP)-framed IPX packet. The subinterfaces can be configured to support multiple Frame Relay PVCs.

In this example, a subinterface is configured for serial line 2, which is configured for Frame Relay encapsulation. The subinterface is called 2.1 to indicate that it is subinterface 1 of serial interface 2.

```
Router(config)# interface serial 2
Router(config-if)# encapsulation frame-relay
Router(config-if)# interface serial 2.1
Router(config-subif)# ?
Interface configuration commands:
apollo                  Apollo interface subcommands
appletalk               Appletalk interface subcommands
bandwidth               Set bandwidth informational parameter
bridge-group            Transparent bridging interface parameters
clns                    CLNS interface subcommands
decnet                  Interface DECnet config commands
delay                   Specify interface throughput delay
description             Interface specific description
exit                    Exit from interface configuration mode
frame-relay             Set frame relay parameters
ip                      Interface Internet Protocol config commands
ipx                     Novell interface subcommands
isis                    IS-IS commands
iso-igrp                ISO-IGRP interface subcommands
no                      Negate a command or set its defaults
ntp                     Configure NTP
shutdown                Shutdown the selected interface
```

Router Configuration Mode

Router configuration commands configure an IP routing protocol.

```
Router(config)# router ?
bgp                     Border Gateway Protocol (BGP)
egp                     Exterior Gateway Protocol (EGP)
igrp                    Interior Gateway Routing Protocol (IGRP)
isis                    ISO IS-IS iso-igrp IGRP for OSI networks
ospf                    Open Shortest Path First (OSPF)
rip                     Routing Information Protocol (RIP)
static                  Static CLNS Routing
```

This example displays how a router is configured to support the Routing Information Protocol (RIP).

```
Router(config)# router rip
Router(config-router)# ?
router configuration commands:
default-information     Control distribution of default information
default-metric          Set metric of redistributed routes
distance                Define an administrative distance
distribute-list         Filter networks in routing updates
exit                    Exit from routing protocol configuration mode
help                    Description of the interactive help system
neighbor                Specify a neighbor router
network                 Enable routing on an IP network
no                      Negate or set default values of a command
offset-list             Add or subtract offset from IGRP, RIP, or HELLO metrics
passive-interface       Suppress routing updates on an interface
redistribute            Redistribute information from another routing protocol
timers                  Adjust routing timers
```

IPX-Router Configuration Mode

Internet Packet Exchange (IPX) is a Novell network-layer protocol. In this example, IPX RIP routing is configured.

```
Router(config)# ipx router rip
Router(config-ipx-router)# ?
Novell router configuration commands:
distribute-list         Filter networks in routing updates
exit                    Exit from IPX routing protocol configuration mode
help                    Description of the interactive help system
network                 Enable routing on an IPX network
no                      Negate or set default values of a command
redistribute            Enable routing protocol redistribution
```

Route-Map Configuration Mode

Use the route-map configuration mode to configure routing table and source and destination information. In this example, a route map named arizona1 is configured.

```
Router(config)# route-map arizona1
Router(config-route-map)# ?
Route Map configuration commands:
exit                    Exit from route-map configuration mode
help                    Description of the interactive help system
match                   Match values from routing table
no                      Negate or set default values of a command
set                     Set values in destination routing protocol
```

Key Chain Configuration Mode

From key chain configuration mode, you can manage authentication keys that routing protocols use. To enter this configuration mode and use Key Chain configuration commands, you must first enable RIP authentication. For more information about enabling RIP, refer to Enable RIP.

Key Chain Key Configuration Mode

Once you define a key chain, use the key chain key configuration mode to configure the keys on the key chain.

Response Time Reporter Configuration Mode

Use the response time reporter feature to monitor network performance, network resources, and applications by measuring response times and availability. With this feature you can perform troubleshooting, problem notifications, and pre-problem analysis based on response time reporter statistics.

Access-List Configuration Mode

All Internet Protocol (IP) access lists can be identified by a number; standard IP access lists are numbered 1 to 99 and extended IP access lists are numbered 100 to 199. Some IP access lists can also be identified by a name. Use access-list configuration mode when you are creating a named IP access list.

In this example, an IP access list named flag is created and the commands available in access-list configuration mode are listed.

```
Router(config)# ip access-list extended flag
Router(config-ext-nacl)# ?
Ext Access List configuration commands:
deny                    Specify packets to reject
dynamic                 Specify a DYNAMIC list of PERMITs or DENYs
exit                    Exit from access-list configuration mode
no                      Negate or set default values of a command
permit                  Specify packets to forward
```

Context-Sensitive Help

The first level of help available with the user interface is context-sensitive help. Entering a question mark (?) at the system prompt displays a list of commands available for each command mode. You can also get a list of any command's associated keywords and arguments with the context-sensitive help feature.

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

Help Command	Command Format
Obtain a brief description of the help system in any command mode.	**`help`**
Receive help for the full set of user-level commands when you type a question mark (?).	**`full-help`**
Receive help for the full set of user-level commands for this exec session.	**`terminal full-help`**
Obtain a list of commands that begin with a particular character string.	***abbreviated-command-entry***?
Complete a partial command name.	***abbreviated-command-entry***<Tab>
List all commands available for a particular command mode.	**?**
List a command's associated keywords.	***command*** ?
List a keyword's associated arguments.	***command keyword*** ?

Warning: When using context-sensitive help, the space (or lack of a space) before the question mark (?) is significant.

Get Word Help

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

Get Command Syntax Help

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

Get Help for Abbreviated Commands

You can abbreviate commands and keywords to the number of characters that allow a unique abbreviation. For example, you can abbreviat the **`show`** command to **`sh`**.

Examples

Enter the **`help`** command, which is available in any command mode, for a brief description of the help system:

```
Router# help
Help may be requested at any point in a command by entering a question mark '?'.
If nothing matches, the help list will be empty and you must back up until
entering a '?' shows the available options. Two styles of help are provided:
1. Full help is available when you are ready to enter a command argument (e.g.
'show ?') and describes each possible argument.
2. Partial help is provided when an abbreviated argument is entered and you want
to know what arguments match the input (e.g. 'show pr?'.)
```

These examples illustrate how the context-sensitive help feature enables you to create an access list from configuration mode.

- Type the letters **`co`** at the system prompt, followed by a question mark (?). Do not leave a space between the last letter and the question mark (?). The system provides the commands that begin with **`co`**.

```
Router# co?
   configure connect copy
```

- Enter the **configure** command followed by a space and a question mark (?) to list the command's keywords and a brief explanation:

```
Router# configure ?
memory                  Configure from NV memory
network                 Configure from a TFTP network host
terminal                Configure from the terminal

<cr>
```

The **<cr>** symbol by itself indicates there are no more keywords or arguments. Press <Enter> to execute the command.

- Enter the **terminal** keyword to enter configuration mode from the terminal:

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

- Enter the **access-list** command followed by a space and question mark (?) to list the command's keywords:

```
Router(config)# access-list ?
<1-99>                  IP standard access list
<100-199>               IP extended access list
<1000-1099>             IPX SAP access list
<1100-1199>             Extended 48-bit MAC address access list
<200-299>               Protocol type-code access list
<300-399>               DECnet access list
<400-499>               XNS standard access list
<500-599>               XNS extended access list
<600-699>               Appletalk access list
<700-799>               48-bit MAC address access list
<800-899>               IPX standard access list
<900-999>               IPX extended access list
```

- Enter the access list number **99**. Then, enter another question mark (?) to see the arguments and brief explanations for the keyword:

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

- Enter the **deny** keyword followed by a question mark (?) to list additional options:

```
Router(config)# access-list 99 deny ?
A.B.C.D                     Address to match
```

- Enter the IP address followed by a question mark (?) to list additional options:

```
Router(config)# access-list 99 deny 131.108.134.0 ?
A.B.C.D                     Mask of bits to ignore
<cr>
```

- Enter the wildcard mask followed by a question mark (?) to list further options.

```
Router(config)# access-list 99 deny 131.108.134.0 0.0.0.255 ?
<cr>
Router(config)# access-list 99 deny 131.108.134.0 0.0.0.255
```

Check Command Syntax

The user interface provides error isolation by using a caret symbol (^ as an indicator. The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument. The error location indicator and interactive help system allow you to find and correct syntax errors easily.

In this example, context-sensitive help is used to check the syntax for setting the clock.

```
Router# clock ?
set                         Set the time and date
Router# clock
```

The help output shows that the set keyword is required. Check the syntax for entering the time:

```
Router# clock set ?
hh:mm:ss                Current time
Router# clock set
Enter the current time:
Router# clock set 13:32:00
% Incomplete command.
```

The system indicates that you need to provide additional arguments to complete the command. Press <Ctrl-P> (refer to Command History Features) to automatically repeat the previous command entry. Then add a space and question mark (?) to reveal the additional arguments:

```
Router# clock set 13:32:00 ?
<1-31>                  Day of the month
January                 Month of the year
February
March
April
May
June
July
August
September
October
November
December
```

Now you can complete the command entry:

```
Router# clock set 13:32:00 23 February 93
                                          ^
% Invalid input detected at '^' marker.
```

The caret symbol **(^)** and help response indicate an error at 93. To list the correct syntax, enter the command up to the point where the erro occurred and then enter a question mark (?):

```
Router# clock set 13:32:00 23 February ?
<1997-2035>             Year
Router# clock set 13:32:00 23 February
```

Enter the year using the correct syntax and press <Enter> to execut the command.

```
Router# clock set 13:32:00 23 February 1997
```

Command History Features

The Cisco IOS software user interface provides a history, or record, o commands that you have entered. This feature is particularly useful fo recalling long or complex commands or entries, including access lists. By default, the system records 10 command lines in its history buffer. The following commands are entered from user EXEC mode.

***Warning:** Many of the commands described in this section refer to arrow keys as well as alternate keystrokes. Please note that arrow keys function only on ANSI-compatible terminals such as VT100s.*

History Commands	Command Format
Set the number of command lines the system will record during the current terminal session.	**`terminal history [size`** ***`number-of-lines`*** **`]`**
Reset the number of lines saved in th history buffer to the default of 10 lines.	**`terminal no history size`**
Recall commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.	Press <Ctrl-P> or the up arrow key
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.	Press <Ctrl-N> or the down arrow key
List the last several commands you entered while in EXEC mode.	**`show history`**
Disable command history during th current terminal session.	**`terminal no history`**

Editing Features

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

Notes:

- ***Many of the commands described in this section refer to arrow keys as well as alternate keystrokes. Please note that arrow keys function only on ANSI-compatible terminals such as VT100s.***
- ***The --More- prompt is used for any output that has more lines than can be displayed on the terminal screen, including `show` command output. You can use the keystrokes listed above whenever you see th --More- prompt.***
- ***You might want to disable enhanced editing if you have prebuilt scripts, such as scripts that do not interact well when enhanced editing is enabled. You can reenable enhanced editing mode with the terminal editing command***

.

Editing Commands	Command Format
In user EXEC mode, reenable the enhanced editing mode for the current terminal session.	`terminal editing`
Move the cursor back one character.	Press <Ctrl-B> or press the left arrow key
Move the cursor forward one character.	Press <Ctrl-F> or press the right arrow key
Move the cursor to the beginning of the command line.	Press <Ctrl-A>
Move the cursor to the end of th command line.	Press <Ctrl-E>
Move the cursor back one word.	Press <Esc-B>
Move the cursor forward one word.	Press <Esc-F>.
Prompt the system to complete a partial entry.	Press <Tab> or <Ctrl-I>
Obtain a list of commands that begin with that set of characters.	`?`
Recall the most recent entry in the buffer.	Press <Ctrl-Y>

Editing Commands (Contd.)	Command Format
Recall the next buffer entry. The buffer contains only the last 10 items you have deleted or cut. If you press <Esc-Y> more than 10 times, you will cycle back to the first buffer entry.	Press <Esc-Y>
Erase the character to the left of the cursor.	Press <Delete> or <Backspace>
Delete the character at the cursor.	Press <Ctrl-D>
Delete all characters from the cursor to the end of the command line.	Press <Ctrl-K>
Delete all characters from the cursor to the beginning of the command line.	Press <Ctrl-U> or <Ctrl-X>
Delete the word to the left of the cursor.	Press <Ctrl-W>
Scroll down one line.	Press <Enter>
Scroll down one screen.	Press the <Space> bar
Redisplay the current command line.	Press <Ctrl-L> or <Ctrl-R>
Transpose mistyped characters. The character to the left of the cursor will be transposed with the characte located at the cursor.	Press <Ctrl-T>
Capitalize at the cursor.	Press <Esc>, then <C>
Change the word at the cursor to low ercase.	Press <Esc>, then <L>
Capitalize letters from the cursor to the end of the word.	Press <Esc>, then <U>
Insert a system code for this purpose.	Press <Ctrl-V> or <Esc-Q>
Disable enhanced editing mode and revert to the editing mode of previous releases.	**`terminal no editing`**

Edit Command Lines that Wrap

The Cisco IOS software assumes that you have a terminal screen that is 80 characters wide. If your terminal displays a different number of characters on each line, use the **`terminal width`** command.

When the cursor reaches the right margin, the command line shifts 10 spaces to the left. You cannot see the first ten characters of the line, but you can scroll back and check the syntax at the beginning of the command. To scroll back, press <Ctrl-B> or the left arrow key repeatedly until you scroll back to the beginning of the command entry, or press <Ctrl-A> to return directly to the beginning of the line.

In this example, the **`access-list`** command entry extends beyond one line. When the cursor first reaches the end of the line, the line is shifted 10 spaces to the left and is redisplayed. The dollar sign ($) in dicates that the line has been scrolled to the left. Each time the cursor reaches the end of the line, the line is again shifted 10 spaces to the left.

```
Router(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108
Router(config)# $ 101 permit tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.2
Router(config)# $t tcp 131.108.2.5 255.255.255.0 131.108.1.20 255.255.255.0
Router(config)# $108.2.5 255.255.255.0 131.108.1.20 255.255.255.0 eq 45
```

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

```
Router(config)# access-list 101 permit tcp 131.108.2.5 255.255.255.0 131.108$
```

Web Browser Interface

You can issue most of the AI2524 commands using a Web browser. This AI2524 feature is accessed by using the Web Browser interface, which is accessed from the router's home page.

From the router's home page, click on the Monitor the Router hyper text link to display a Web page with a Command field. You can typ commands in this field as if you were entering commands at a terminal connected to the router. The page also displays a list of commands (hypertext links) you can execute by clicking.

Web Browser Interface Task List

To use the Web Browser interface to issue commands, perform the tasks in the following list:

- Enable the Web Browser Interface
- Use the Correct Hardware and Software
- Access Your Router's Home Pag
- Issue Commands Using the Web Browser Interfac
- Enter Commands Using Hypertext Links
- Enter Commands Using the Command Field
- Enter Commands Using the URL Window

Enable the Web Browser Interface

To enable your router to be configured from a browser using the Web Browser interface, type this command in global configuration mode:

```
ip http server
```

Once the Web Browser interface is enabled, you can issue AI2524 commands to your router using a Web browser.

Use Compatible Hardware and Software

To use the Web Browser interface, your computer must have a World Wide Web browser. The Web Browser interface works with most browsers, including Netscape Navigator. Your Web browser must be able to read and submit forms. The earliest versions of Mosaic might have problems using the Web Browser interface, because they either cannot submit forms or have difficulty doing so.

The computer must be connected to the same network as the router or access server.

Access Your Router's Home Page

Perform these steps to access the home page for your router or access server:

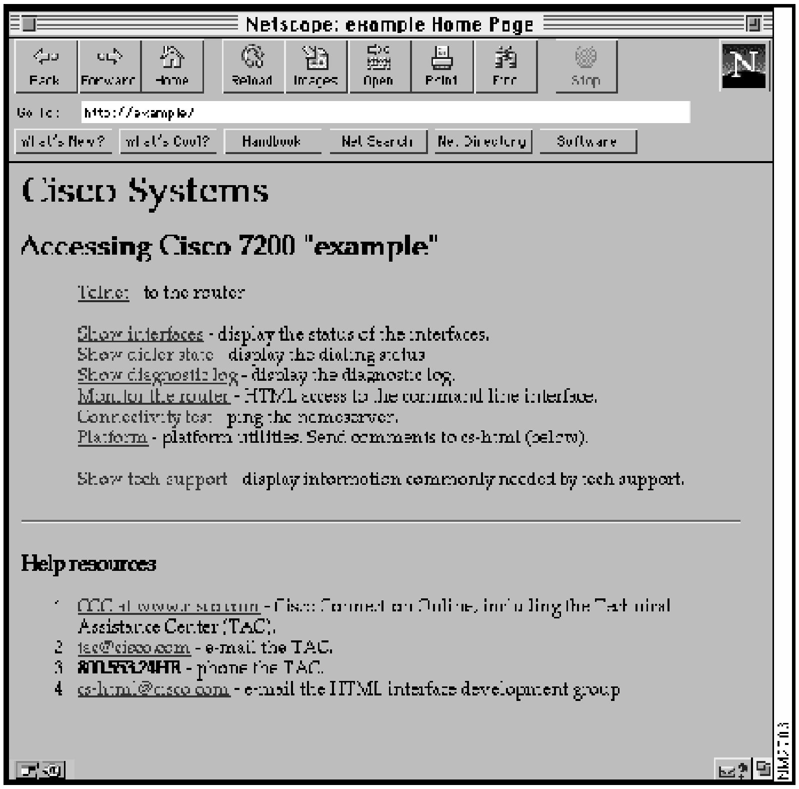
1. Enter the name of the router or access server in the URL field of your Web browser and press <Enter>. The browser prompts you for the password for the router or access server.

2. Enter the password.

Warning: ***The name and password for your router and access server are designated as part of the configuration process. Contact your network administrator if you do not have this information.***

The browser should display the home page for your router or access server.The router's home page looks something like the home page shown in Figure 4-1.

Figure 4-1:Example of a Home Page



Cisco Systems

Accessing Cisco 7200 "example"

Telnet - to the router

Show interfaces - display the status of the interfaces.
Show dialer state - display the dialing status
Show diagnostic log - display the diagnostic log.
Monitor the router - HTML access to the command line interface.
Connectivity test - ping the nameserver.
Platform - platform utilities. Send comments to cs-html (below).

Show tech support - display information commonly needed by tech support.

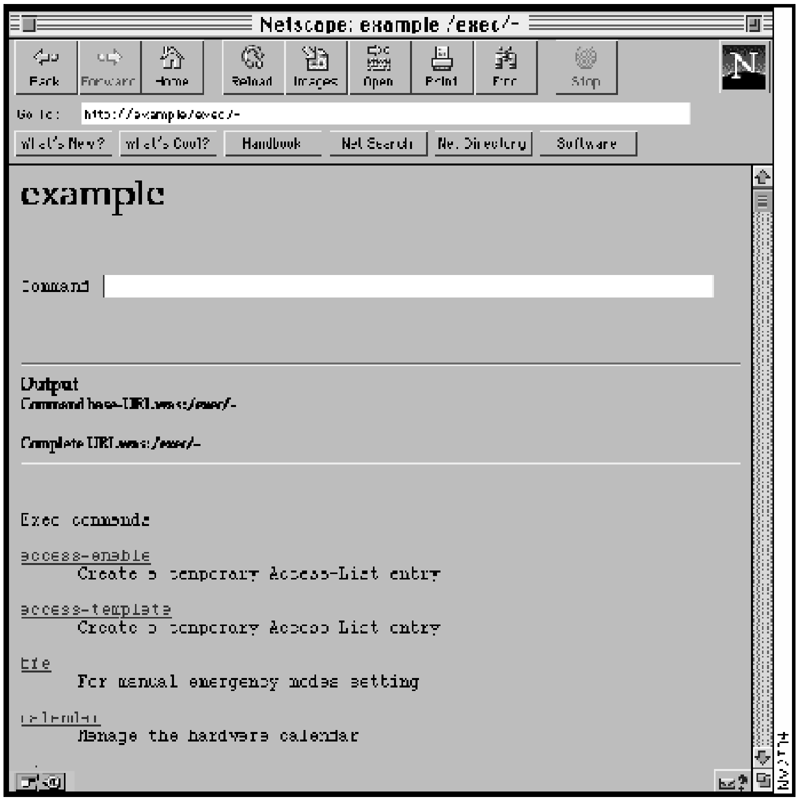
Help resources

1 CCO at www.cisco.com - Cisco Connection Online, including the Technical Assistance Center (TAC).
2 tac@cisco.com - e-mail the TAC.
3 800.553.24HR - phone the TAC.
4 cs-html@cisco.com - e-mail the HTML interface development group

Issue Commands Using the Web Browser Interface

To issue commands using the Web Browser interface, click Monitor the router in the first list of hypertext links on the home page. This displays the Web page shown in Figure 4-2.

Figure 4-2:The Command Field Web Page for a Router Named example



http://example/exec/-

example

Command

Output
Command base-URL was: /exec/-

Complete URL was: /exec/-

```
Exec commands

access-enable
        Create a temporary Access-List entry

access-template
        Create a temporary Access List entry

bfe
        For manual emergency modes setting

calendar
        Manage the hardware calendar
```

Enter Commands Using Hypertext Links

To enter a command using hypertext links, scroll through the commands listed at the bottom of the screen and click the one you want to execute. If the link is a complete command, it is executed. If the command has more parameters, another list of command hypertext links is displayed. Scroll through this second list and click the one you want to execute.

If the command is a request for information, like **show** command, the information is displayed in the Web browser window.

If the command requires a variable, a form in which you can enter the variable is displayed.

Enter Commands Using the Command Field

Entering the command in the command field is just like entering it at a terminal console. If you are uncertain of the options available for particular command, type a question mark (?).

For example, typing **`show ?`** in the command field displays the parameters for the **`show`** command. The Web Browser interface displays the parameters as hypertext links. To select a parameter, click one of the links or enter the parameter in the command field.

Enter Commands Using the URL Window

You can issue a command using the Universal Resource Locator (URL) window for the Web browser.

For example, to execute a **`show configuration`** command on a router named example, you would enter this in the URL window:

```
http://example/exec/show/configuration
```

The Web browser then displays the configuration for the exampl router. To save effort, modify the URL in the URL window in the browser control bar instead of retyping the entire URL.

The difference between entering a command in the command field and entering a command in the URL window is that in the URL window, command modes and options should be separated by slashes, not spaces.

Chapter 5: Using AutoInstall

Introduction

This chapter provides information about AutoInstall, a procedure that allows you to configure a new router automatically and dynamically. The AutoInstall procedure involves connecting a new router to a network where an existing router is preconfigured, turning on the new router, and enabling it with a configuration file that is automatically downloaded from a Trivial File Transfer Protocol (TFTP) server.

Preparing for AutoInstall

Take these steps to prepare your router for the AutoInstall process:

1. Attach the WAN cable to the router.
2. Turn ON power to the router.

   The router will load the operating system image from Flash memory. If the remote end of the WAN connection is connected and properly configured, the AutoInstall process will begin.

   If AutoInstall successfully completes, you can write the configuration data to the router's NVRAM. Perform the following step to complete this task.

3. Enter the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
```

Taking this step saves the configuration settings that the AutoIn stall process created in the router. If you do not do this, your con figuration will be lost the next time you reload the router.

The next sections provide AutoInstall requirements and an overview of its procedure. For information about starting AutoInstall, refer to Perform the AutoInstall Procedur .

AutoInstall Requirements

The AutoInstall process is designed to configure the router automatically after connection to your WAN. In order for AutoInstall to work properly, a Transmission Control Protocol/Internet Protocol (TCP/IP) host on your network must be preconfigured to provide the required configuration files. The TCP/IP host may exist anywhere on the network as long as these two conditions are maintained:

- The host must be on the remote side of the router's synchronous serial connection to the WAN.
- User Datagram Protocol (UDP) broadcasts to and from the router and the TCP/IP host must be enabled.

  This functionality is coordinated by your system administrator at the site where the TCP/IP host is located. You should not attempt to use AutoInstall unless the required files have been provided on the TCP/IP host.

Note: ***AutoInstall works on synchronous serial connections only. The 2-wire switched 56-kbps DSU/CSU module op erates on switched 56-kbps circuits only; therefore, you cannot use it for AutoInstall.***

In addition, your system must meet these requirements:

- Routers must be physically attached to the network using one o more of the following interface types: Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), serial with High-Level Dat Link Control (HDLC) encapsulation, or serial with Frame Relay encapsulation. HDLC is the default serial encapsulation. If the AutoInstall process fails over HDLC, the Cisco IOS software automatically configures Frame Relay encapsulation.

Note: ***For Token Ring interfaces, only those that set ring speed with physical jumpers support AutoInstall. AutoInstall does not work with Token Ring interfaces for which the ring speed must be set with software configuration commands. If the ring speed is not set, the interface is set to shutdown mode.***

- You must complete Step 1 and either Step 2 or 3:

1. A configuration file for the new router must reside on a TFTP server. This file can contain the minimum or full configuration needed for the administrator to Telnet into the new router for configuration. In addition, make sure to complete one of the next two procedures:
2. A file named network-confg also must reside on the server. The file must have an Internet Protocol (IP) host name entry for the new router. The server must be reachable from the existing router.

3. An IP address-to-host name mapping for the new router must be added to a Domain Naming System (DNS) database file.

- If the existing router is to help automatically install the new route via an HDLC-encapsulated serial interface using Serial Line Address Resolution Protocol (SLARP), that interface must be configured with an IP address whose host portion has the value 1 or 2. (AutoInstall over Frame Relay does not have this address constraint.) Subnet masks of any size are supported.

- If the existing router is to help automatically install the new route using a Frame Relay-encapsulated serial interface, that interface must be configured with:

  An IP helper address pointing to the TFTP server. In this example, 171.69.2.75 is the address of the TFTP server:

```
ip helper 171.69.2.75
```

A Frame Relay map pointing back to the new router. In this example, 172.21.177.100 is the IP address of the new router's serial interface, and 100 is the PVC identifier:

```
frame-relay map ip 172.21.177.100 100 dlci
```

- If the existing router is to help automatically install the new route via an Ethernet, Token Ring, or FDDI interface using BOOTP o Reverse Address Resolution Protocol (RARP), a BOOTP or RARP server also must be set up to map the new router's Media Access Control (MAC) address to its IP address.

- IP helper addresses should be configured to forward the TFTP and DNS broadcast requests from the new router to the host providing those services.

Use a DOS-Based TFTP Server

AutoInstall over Frame Relay and other WAN encapsulations support downloading configuration files from UNIX-based and DOS-based TFTP servers. Other booting mechanisms such as RARP and SLARP also support UNIX-based and DOS-based TFTP servers.

The DOS format of the UNIX network-confg file that must reside on the server must be eight characters or fewer, with a three-letter exten sion. Therefore, when an attempt to load network-confg fails, AutoInstall automatically attempts to download the filecisconet.cfg from the TFTP server.

If cisconet.cfg exists and is downloaded successfully, the server is assumed to be a DOS machine. The AutoInstall program then attempts to resolve the host name for the router through host commands in cisconet.cfg.

If cisconet.cfg does not exist or cannot be downloaded, or if the program is unable to resolve a host name, DNS attempts to resolve the host name. If DNS cannot resolve the host name, the router attempts to download ciscortr.cfg. If the host name is longer than eight characters, it is truncated to eight characters. For example, a router with a host name australia will be treated as australi and AutoInstall will attempt to download australi.cfg.

The format of cisconet.cfg and ciscortr.cfg is the same as those described for network-confg and hostname-confg.

If neither network-confg nor cisconet.cfg exists and DNS is unable to resolve the host name, AutoInstall attempts to load router-confg, and then ciscortr.cfg if router-confg does not exist or cannot be downloaded. The cycle is repeated three times.

How AutoInstall Works

Once the requirements for using AutoInstall are met, the dynamic configuration of the new router occurs in this order:

1. The new router acquires its IP address. Depending on the interface connection between the two routers and/or access servers, the new router's IP address is dynamically resolved by either SLARP requests or BOOTP or RARP requests.

2. The new router resolves its name through network-confg, cisconet.cfg, or DNS.

3. The new router automatically requests and downloads its configuration file from a TFTP server.

4. If a host name is not resolved, the new router attempts to load router-confg or ciscortr.cfg.

Acquire the New Router's IP Address

The new router (newrouter) resolves its interface's IP addresses as:

- If newrouter is connected by an HDLC-encapsulated serial line to the existing router (existing), newrouter sends a SLARP request to existing.

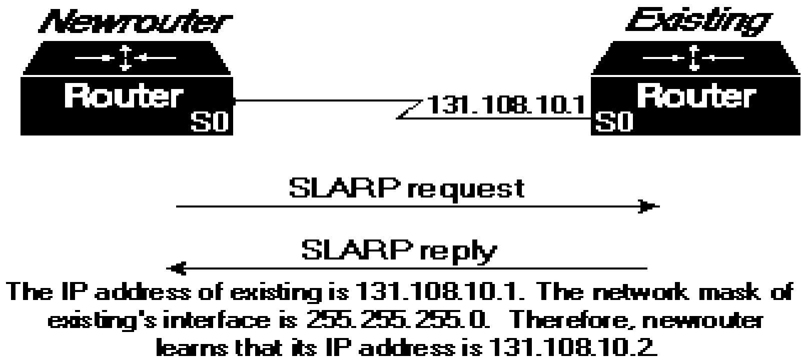
- If newrouter is connected by an Ethernet, Token Ring, or FDDI interface, it broadcasts BOOTP and RARP requests.

- If newrouter is connected by a Frame Relay-encapsulated serial interface, it first attempts the HDLC automatic installation process and then attempts the BOOTP or RARP process over Ethernet, Token Ring, or FDDI. If both attempts fail, the new router attempts to automatically install over Frame Relay. In this case, a BOOTP request is sent over the lowest numbered serial or HSSI interface.

The existing router (existing) responds in one of these ways depending on the request type:

- In response to a SLARP request, existing sends a SLARP reply packet to newrouter. The reply packet contains the IP address and netmask of existing. If the host portion of the IP address in the SLARP response is 1, newrouter configures its interface using the value 2 as the host portion of its IP address and vice versa. (See Figure 5-1.)

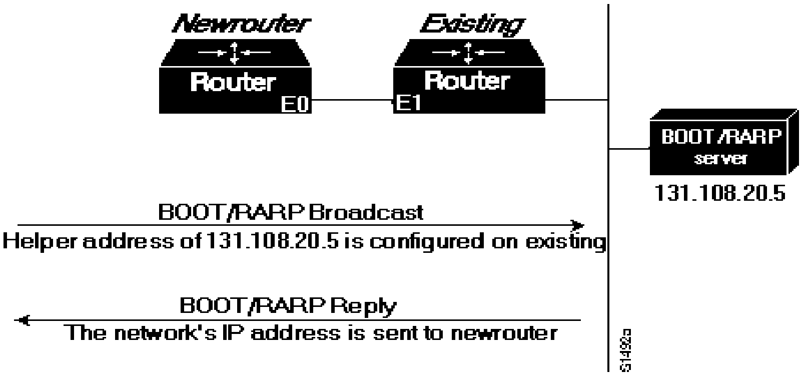
Figure 5-1:Using SLARP to Acquire the New Router's IP Address


Newrouter
Router
S0
131.108.10.1
Existing
Router
S0
SLARP request
SLARP reply
The IP address of existing is 131.108.10.1. The network mask of existing's interface is 255.255.255.0. Therefore, newrouter learns that its IP address is 131.108.10.2

- In response to BOOTP or RARP requests, an IP address is sent from the BOOTP or RARP server to newrouter.

  A BOOTP or RARP server must have already been set up to map newrouter's MAC address to its IP address. If the BOOTP server does not reside on the directly attached network segment, routers between newrouter and the BOOTP server can be configured with the **`ip helper-address`** command to allow the request and response to be forwarded between segments, as shown in Figur 5-2.

Figure 5-2:Use BOOTP or RARP to Acquire the New Router's IP Address


Newrouter
Router
E0
Existing
Router
E1
BOOT/RARP server
131.108.20.5
BOOT/RARP Broadcast
Helper address of 131.108.20.5 is configured on existing
BOOT/RARP Reply
The network's IP address is sent to newrouter
S1492a

AutoInstall over Frame Relay is a special case in that the existing router acts as a BOOTP server and responds to the incoming BOOTP request. Only a helper address and a Frame Relay map need to be set up. No MAC-to-IP address map is needed on the existing router.

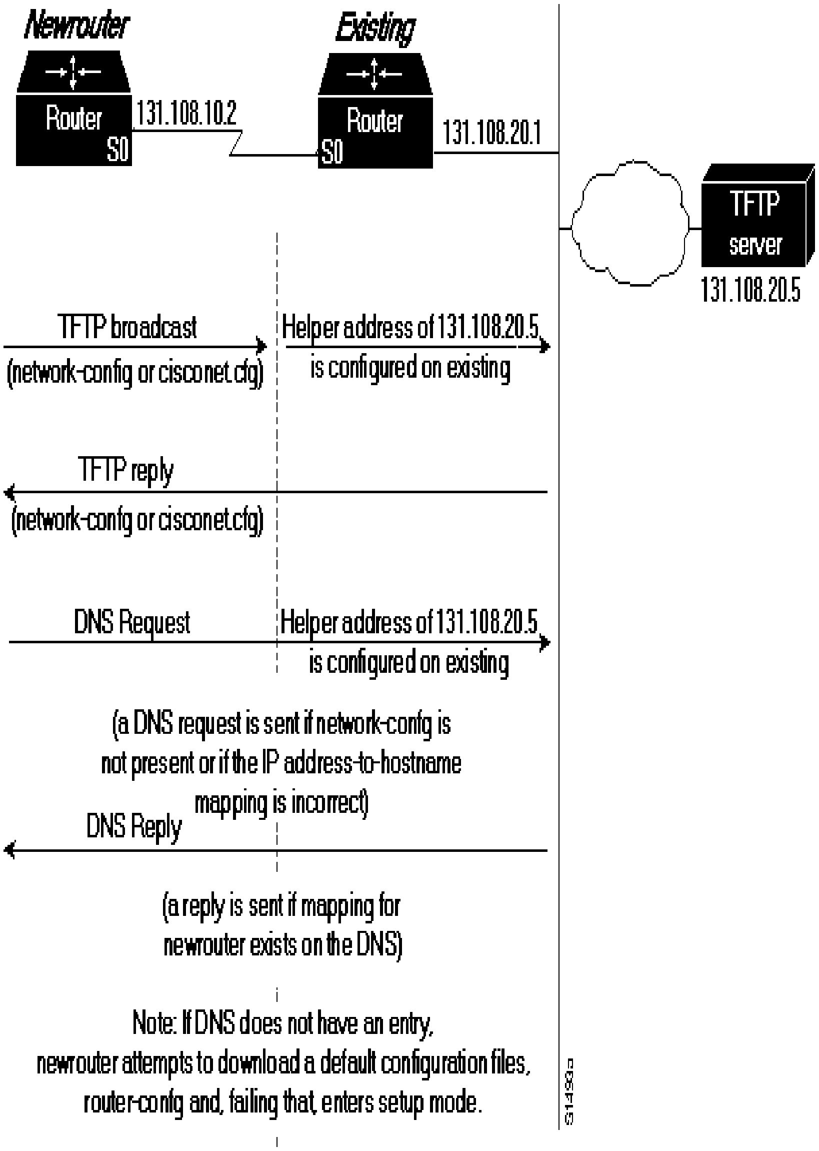
The AI2524 routers can be configured to act as a RARP server.

Because the router attempts to resolve its host name as soon as one interface resolves its IP address, only one IP address needs to be set up with SLARP, BOOTP, or RARP.

Resolve the IP Address to the Host Name

The new router resolves its IP address-to-host name mapping by sending a TFTP broadcast requesting the file network-confg, as shown in Figure 5-3.

Figure 5-3:Dynamically Resolve the New Router's IP Address-to-Host Name Mapping


Newrouter
Existing
Router
S0
131.108.10.2
Router
S0
131.108.20.1
TFTP server
131.108.20.5
TFTP broadcast
(network-config or cisconet.cfg)
Helper address of 131.108.20.5
is configured on existing
TFTP reply
(network-confg or cisconet.cfg)
DNS Request
Helper address of 131.108.20.5
is configured on existing
(a DNS request is sent if network-confg is
not present or if the IP address-to-hostname
mapping is incorrect)
DNS Reply
(a reply is sent if mapping for
newrouter exists on the DNS)
Note: If DNS does not have an entry,
newrouter attempts to download a default configuration files,
router-confg and, failing that, enters setup mode.
S1493a

The network-confg file is a configuration file generally shared by several routers. In this case, it maps the IP address of the new router (just obtained dynamically) to the name of the new router. The file network-confg must reside on a reachable TFTP server and must be globally readable.

This is an example of a minimal network-confg file that maps the IP address of the new router (131.108.10.2) to the name newrouter. The address of the new router was learned via SLARP and is based on *existing*'s IP address of 131.108.10.1.

```
ip host newrouter 131.108.10.2
```

If you are not using AutoInstall over Frame Relay, the host portion of the address must be 1 or 2. AutoInstall over Frame Relay does not hav this addressing constraint.

If newrouter does not receive a network-confg or a cisconet.cfg file, or if the IP address-to-host-name mapping does not match the newly acquired IP address, newrouter sends a DNS broadcast. If DNS is configured and has an entry that maps newrouter's SLARP, BOOTP, or RARP-acquired IP address to its name, newrouter successfully resolves its name.

If DNS does not have an entry that maps the new router's SLARP, BOOTP, or RARP-acquired address to its name, the new router cannot resolve its host name. The new router attempts to download a default configuration file as described in the next section, and failing that, enters setup mode or enters user EXEC mode with AutoInstall over Frame Relay.

Download the New Router's Host Configuration File

After the router successfully resolves its host name, newrouter sends TFTP broadcast requesting the file newrouter-confg or newrouter.cfg. The name newrouter-confg must be in all lowercase letters, even if th true host name is not. If newrouter cannot resolve its host name, it sends a TFTP broadcast requesting the default host configuration fil router-confg. The file is downloaded to newrouter, where the configuration commands take effect immediately.

When using AutoInstall over Frame Relay, you are put into setup mode while the AutoInstall process is running. If the configuration file is successfully installed, the setup process is terminated. If you expect the AutoInstall process to be successful, either do not respond to the setup prompts or respond to the prompts as:

```
Would you like to enter the initial configuration dialog? [yes]: no
Would you like to terminate autoinstall? [yes]: no
```

If you do not expect the AutoInstall process to be successful, create configuration file by responding to the setup prompts. The AutoInstall process is terminated transparently.

You will see this display as the AutoInstall operation is in progress:

```
Please Wait. AutoInstall being attempted!!!!!!!!!!!!!
```

If the host configuration file contains only the minimal information, you must connect to existing using Telnet. From there, connect via Telnet to newrouter. Then, run the **setup** command to configure ne wrouter. Refer to Use Setup for Configuration Changes, for specific information about the **setup** command.

If the host configuration file is complete, newrouter should be fully operational. You can enter the **enable** command (with the system administrator password) at the system prompt on newrouter. Then, enter the **copy running-config startup-config** command to save the information in the recently obtained configuration file into nonvolatile random-access memory (NVRAM) or to the location specified by the CONFIG_FILE environment variable. If it must reload, newrouter simply loads its configuration file from NVRAM.

If the TFTP request fails, or if newrouter still has not obtained the IP addresses of all its interfaces, and if those addresses are not contained in the host configuration file, then newrouter enters setup mode automatically. Setup mode prompts you for manual configuration of the Cisco IOS software at the console. The new router continues to issue broadcasts to attempt to learn its host name and obtain any unresolved interface addresses. The broadcast frequency will dwindle to every 10 minutes after several attempts. Refer to Use Setup for Configuration Change , for specific information about th **setup** command.

Perform the AutoInstall Procedure

To dynamically configure a new router using AutoInstall, complete these tasks. Steps 1, 2, and 3 are completed by the central administrator. Step 4 is completed by the person at the remote site.

1. Modify the existing router's configuration to support AutoInstall.
2. Set up the TFTP server to support AutoInstall.
3. Set up the BOOTP or RARP server if needed. A BOOTP or RARP server is required for AutoInstall using an Ethernet, Token Ring, FDDI, or Frame Relay-encapsulated serial interface. With a Frame Relay-encapsulated serial interface, the existing router acts as the BOOTP server. A BOOTP or RARP server is not required for AutoInstall using an HDLC-encapsulated serial interface.
4. Connect the new router to the network.

Modify the Existing Router's Configuration

You can use any of these interfaces:

- An HDLC-encapsulated serial line (the default configuration for serial line)
- An Ethernet, Token Ring, FDDI interface
- A Frame Relay-encapsulated serial line

Use an HDLC-Encapsulated Serial Interface Connection

To set up AutoInstall via a serial line with HDLC encapsulation (the default), you must configure the existing router. Perform these steps, beginning in global configuration mode:

1. Configure the serial interface that connects to the new router with HDLC encapsulation (the default), and enter interface configura tion mode.

   ```
   interface serial interface-number
   ```

2. Enter an IP address for the interface. The host portion of the ad dress must have a value of 1 or 2. (AutoInstall over Frame Relay does not have this address constraint.

   ```
    ip address address mask
   ```

3. Configure a helper address for the serial interface to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.

   ```
   ip helper-address address
   ```

4. Optionally, configure a DCE clock rate for the serial line, unless an external clock is being used. This step is needed only for DCE appliques.

   clock rate _**bps**_

5. To exit configuration mode, press <Ctrl-Z>.

6. Save the configuration file to your startup configuration.

   copy running-config startup-config

In this example, the existing router's configuration file contains the commands needed to configure the router for AutoInstall on a serial line using HDLC encapsulation:

```
Router# configure terminal
interface serial 0
  ip address 172.31.10.1 255.255.255.0
  ip helper-address 172.31.20.5
  Ctrl-Z
Router(config)# copy running-config startup-config
```

Use an Ethernet, Token Ring, or FDDI Interface Connection

To set up AutoInstall using an Ethernet, Token Ring, or FDDI interface, you must modify the configuration of the existing router. Perform these steps, beginning in global configuration mode:

1. Configure a LAN interface, and enter interface configuration mode.

   interface {_**ethernet |tokenring | fddi}**_
   **interface-number**

2. Enter an IP address for the interface.

   ip address _**address mask**_

3. Optionally, configure a helper address to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.

   ip helper-address _**address2**_

4. To exit configuration mode, press <Ctrl-Z>.

5. Save the configuration file to your startup configuration. This step saves the configuration to NVRAM.

   copy running-config startup-config

Typically, the local-area network (LAN) interface and IP address ar already configured on the existing router. You might need to configur an IP helper address if the TFTP server is not on the same network as the new router.

In this example, the existing router's configuration file contains the commands needed to configure the router for AutoInstall on an Ethernet interface:

```
Router# configure terminal
interface Ethernet 0
   ip address 172.31.10.1 255.255.255.0
   ip helper-address 172.31.20.5
   Ctrl-Z
Router(config)# copy running-config startup-config
```

Use a Frame Relay-Encapsulated Serial Interface Connection

To set up AutoInstall via a serial line with Frame Relay encapsulation, you must configure the existing router. Perform these tasks, beginning in global configuration mode:

1. Configure the serial interface that connects to the new router, and enter interface configuration mode.

   interface serial 0

2. Configure Frame Relay encapsulation on the interface that connects to the new router.

   encapsulation frame-relay

3. Create a Frame Relay map pointing back to the new router, or point-to-point subinterfaces, assign a data link connection identifier (DLCI) to the interface that connects to the new router, and provide the IP address of the serial port on the new router.

   frame-relay map ip ***ip-address dlci***

or

frame-relay interface-dlci ***dlci option*** **[protocol ip** ***ip-address*****]**

4. Enter an IP address for the interface. This step sets the IP address of the existing router.

   ```
   ip address address mask
   ```

5. Configure a helper address for the TFTP server.

   ```
   ip helper-address address3
   ```

6. Optionally, configure a DCE clock rate for the serial line, unless an external clock is being used. This step is needed only for DCE applications.

   ```
   clock rate bps1
   ```

7. To exit configuration mode, press <Ctrl-Z>.

8. Save the configuration file to your startup configuration. This step saves the configuration to NVRAM.

   ```
   copy running-config startup-config
   ```

You must use a DTE interface on the new router because the network always provides the clock signal.

In this example, the existing router's configuration file contains the commands needed to configure the router for Frame Relay AutoInstall on a serial line:

```
Router# configure terminal
interface serial 0
   ip address 172.31.20.20 255.255.255.0
   encapsulation frame-relay
   frame-relay map ip 172.31.10.1 255.255.255.0 48
ip helper-address 172.31.20.5
```

Set Up the TFTP Server

For AutoInstall to work correctly, the new router must resolve its host name and then download a name-confg or a name.cfg file from a TFTP server. The new router can resolve its host name by using a network-confg or a cisconet.cfg file downloaded from a TFTP server or by using the DNS.

To set up a TFTP server to support AutoInstall, complete these tasks. Step 2 includes two ways to resolve the new router's host name. Use the first method if you want to us network-confg file to resolve the new router's host name. Use the second method if you want to us DNS to resolve the new router's host name.

1. Enable TFTP on a server. Consult your host vendor's TFTP serve documentation and RFCs 906 and 783.

2. If you want to use a network-confg or cisconet.cfg file to resolve the new router's name, create the network-confg or cisconet.cfg file containing an IP address-to-host name mapping for the new router.

   Enter the **`ip host`** command into the TFTP config file, not into the router. The IP address must match the IP address that is to be dynamically obtained by the new router, or if you want to use DNS to resolve the new router's name, create an address-to-name mapping entry for the new router in the DNS database. The IP address must match the IP address that is to be dynamically obtained by the new router.

   ```
   ip host hostname address
   ```

   Contact the DNS administrator or refer to RFCs 1101 and 1183.

3. Create th name-confg or name.cfg file, which should reside in th tftpboot directory on the TFTP server. The name part of name-confg or name.cfg filename must match the host name you assigned for the new router in the previous step. Enter configuration commands for the new router into this file.

   The name-confg or the name.cfg file can contain either the new router's full configuration or a minimal configuration.

The minimal configuration file is a virtual terminal password and an enable password. It allows an administrator to gain access (via Telnet into the new router to configure it. If you are using BOOTP or RARP to resolve the address of the new router, the minimal configuration file must also include the IP address to be obtained dynamically using BOOTP or RARP.

You can use the copy running-config tftp command to help you generate the configuration file that you will download during the AutoInstall process.

Note: ***The existing router might need to forward TFTP requests and response packets if the TFTP server is not on the same network segment as the new router. When you modified the existing router's configuration, you specified an IP helper address for this purpose.***

You can save a minimal configuration under a generic newrouter-confg file. Use the **`ip host`** command in the network-confg or cisconet.cfg file to specify newrouter as the host name with the address you dynamically resolve. The new router should then resolve its IP ad dress, host name, and minimal configuration automatically. Use Telnet to connect to the new router from the existing router and use the setup facility to configure the rest of the interfaces. For example, th line in the network-confg or cisconet.cfg file could be similar to:

```
ip host newrouter 131.108.170.1
```

This host configuration file contains the minimal set of commands needed for AutoInstall using SLARP or BOOTP:

```
enable-password letmein
!
line vty 0
password letmein
!
end
```

The preceding example shows a minimal configuration for connecting from a router one hop away. From this configuration, use the setup facility to configure the rest of the interfaces. If the router is more than one hop away, you also must include routing information in the minimal configuration.

This example minimal network configuration file maps the new router's IP address, 131.108.10.2, to the host name *newrouter*. The new router's address was learned via SLARP and is based on the existing router's IP address of 131.108.10.1.

```
ip host newrouter 131.108.10.2
```

Set Up the BOOTP or RARP Server

If the new router is connected to the existing router using an Ethernet, Token Ring, or FDDI interface, you must configure a BOOTP or RARP server to map the new router's MAC address to its IP address. If the new router is connected to the existing router using a serial line with HDLC encapsulation or if you are configuring AutoInstall over Frame Relay, these tasks are not required.

To configure a BOOTP or RARP server, complete one of these tasks:

1. If BOOTP is to be used to resolve the new router's IP address, configure your BOOTP server. Refer to your host vendor's manual pages and to RFCs 951 and 1395.

2. If RARP is to be used to resolve the new router's IP address, configure your RARP server. Refer to your host vendor's manual pages and to RFC 903.

This example host configuration file contains the minimum set of commands needed for AutoInstall using RARP. It includes the IP address that will be obtained dynamically via BOOTP or RARP during the AutoInstall process. When RARP is used, this extra information is needed to specify the proper netmask for the interface.

```
interface ethernet 0
ip address 131.108.10.2 255.255.255.0
enable-password letmein
!
line vty 0
password letmein
!
end
```

Connect the New Router to the Network

Connect the new router to the network using either an HDLC-encapsulated or Frame Relay-encapsulated serial interface or an Ethernet, Token Ring, or FDDI interface. After the router successfully resolves its host name, newrouter sends a TFTP broadcast requesting the file name-confg or name.cfg. The router name must be in all lowercase, even if the true host name is not. The file is downloaded to the new router, where the configuration commands take effect immediately. If the configuration file is complete, the new router should be fully operational. To save the complete configuration to NVRAM, complete these tasks in privileged EXEC mode:

1. Enter privileged mode at the system prompt on the new router.

   `enable password`

2. Save the information from the name-config file into your startup configuration. This step saves the configuration to NVRAM.

   `copy running-config startup-config`

Note: ***Verify that the existing and new routers and/or access servers are connected before entering th*** **`copy running-config startup-config`** ***EXEC command to save configuration changes. Use the*** **`ping`** ***EXEC command to verify connectivity. If an incorrect configuration file is downloaded, the new router will load NVRAM configuration information before it can enter AutoInstall mode.***

If the configuration file is a minimal configuration file, the new router comes up, but with only one interface operational. Complete the following steps to connect to the new router and configure it:

1. Establish a Telnet connection to the existing router.

   ```
   telnet existing
   ```

2. From the existing router, establish a Telnet connection to the new router.

   ```
   telnet newrouter
   ```

3. Enter privileged EXEC mode.

   ```
   enable password
   ```

4. Enter setup mode to configure the new router.

   ```
   setup
   ```

Use Setup for Configuration Changes

The setup command facility is an interactive facility that allows you to perform first-time configuration and other basic configuration procedures on all routers. The facility prompts you to enter basic information needed to start a router quickly and uneventfully.

Although the setup command facility is a quick way to set up a router, you can also use it after first-time startup to perform basic configuration changes. This section focuses on:

- Using the setup command facility after first-time startup
- Using the streamlined setup facility

Refer to your hardware platform's user guide for more information on using setup for first-time startup.

Whenever you use the setup command facility, be sure that you know:

- Router interfaces
- Router protocols
- Bridging setting
- Network addresses for the protocols being configured
- Password strategy for your environment

Setup Command Facility Task List

You can perform these tasks to make configuration changes using the setup command facility. Both tasks are optional.

- Use Setup after First-Time Startup
- Use the Streamlined Setup Facility

Use Setup after First-Time Startup

The command parser allows you to make very detailed changes to your configurations; however, some major configuration changes do not require the granularity provided by the command parser. In these cases, you can use the setup command facility to make major enhancements to your configurations. For example, you might want to use setup to add a protocol suite, to make major addressing scheme changes, or to configure a newly installed interface. Although you can use the command parser to make these major changes, the setup command facility provides you with a high-level view of the configuration and guides you through the configuration change process.

Additionally, if you are not familiar with the command parser, the setup command facility is a particularly valuable tool because it asks you the questions required to make configuration changes.

Note: ***If you use setup to modify a configuration because you have added or modified the hardware, be sure to verify the physical connections using the*** `show version` ***command. Also, verify the logical port assignments using the*** `show running-config` ***command to ensure that you configure the proper port. Refer to your platform's hardware publications for details on physical and logical port assignments.***

To enter the setup command facility, type this command in privileged EXEC mode:

```
setup
```

When you enter the setup command facility after first-time startup, an interactive dialog called the System Configuration Dialog appears on the system console screen. The System Configuration Dialog guides you through the configuration process. It prompts you first for global parameters and then for interface parameters. The values shown in brackets next to each prompt are the default values last set using either the setup command facility or the **`configure`** command.

Note: ***The prompts and the order in which they appear on the screen vary depending on the platform and the interfaces installed in the device.***

You must run through the entire System Configuration Dialog until you find the item you intend to change. To accept default settings fo items you do not want to change, press <Enter>.

To return to the privileged EXEC prompt without making changes and without running through the entire System Configuration Dialog, press <Ctrl-C>.

The facility also provides help text for each prompt. To access help text, press the question mark (?) key at a prompt.

When you complete your changes, the setup command facility shows you the configuration command script created during the setup session. It also asks you if you want to use this configuration. If you answer Yes, the configuration is saved to NVRAM. If you answer No, the configuration is not saved and the process begins again. There is no default for this prompt; you must answer either Yes or No.

Note: ***If any problems exist with the configuration file pointed to in NVRAM, or if the ignore NVRAM bit is set in th configuration register, the router enters the streamlined setup command facility. Refer to Use the Streamlined Setup Facility, for more information.***

This example shows how to use the setup command facility to configure interface serial 0 and to add ARAP and IP/IPX PPP support on the asynchronous interfaces:

```
Router# setup

       ---System Configuration Dialog---

At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Continue with configuration dialog? [yes]:

First, would you like to see the current interface summary? [yes]:
Interface            IP-Address    OK?       Method        Status        Protocol
Ethernet0            172.16.72.2   YES       manual        up            up
Serial0              unassigned    YES       not set       down          down
Serial1              172.16.72.2   YES       not set       up            up

Configuring global parameters:

   Enter host name [Router]:

The enable secret is a one-way cryptographic secret used instead of the enable
password when it exists.

   Enter enable secret [<Use current secret>]:

The enable password is used when there is no enable secret and when using older
software and some boot images.

   Enter enable password [ww]:

Enter virtual terminal password [ww]:
Configure SNMP Network Management? [yes]:
   Community string [public]:
Configure DECnet? [no]:
Configure AppleTalk? [yes]:
   Multizone networks? [no]: yes
Configure IPX? [yes]:
Configure IP? [yes]:
   Configure IGRP routing? [yes]:
     Your IGRP autonomous system number [15]:
```

```
Configure Async lines? [yes]:
   Async line speed [9600]: 57600
   Configure for HW flow control? [yes]:
   Configure for modems? [yes/no]: yes
     Configure for default chat script? [yes]: no
   Configure for Dial-in IP SLIP/PPP access? [no]: yes
     Configure for Dynamic IP addresses? [yes]: no
     Configure Default IP addresses? [no]: yes
     Configure for TCP Header Compression? [yes]: no
     Configure for routing updates on async links? [no]:
   Configure for Async IPX? [yes]:
   Configure for Appletalk Remote Access? [yes]:
     AppleTalk Network for ARAP clients [1]: 20
     Zone name for ARAP clients [ARA Dialins]:

Configuring interface parameters:

Configuring interface Ethernet0:
   Is this interface in use? [yes]:
   Configure IP on this interface? [yes]:
     IP address for this interface [172.16.72.2]:
     Number of bits in subnet field [8]:
     Class B network is 172.16.0.0, 8 subnet bits; mask is /24
   Configure AppleTalk on this interface? [yes]:
     Extended AppleTalk network? [yes]:
     AppleTalk starting cable range [1]:
     AppleTalk ending cable range [1]:
     AppleTalk zone name [Sales]:
     AppleTalk additional zone name:
   Configure IPX on this interface? [yes]:
     IPX network number [1]:

Configuring interface Serial0:
   Is this interface in use? [no]: yes
   Configure IP on this interface? [no]: yes
   Configure IP unnumbered on this interface? [no]: yes
     Assign to which interface [Ethernet0]:
   Configure AppleTalk on this interface? [no]: yes
     Extended AppleTalk network? [yes]:
     AppleTalk starting cable range [2]: 3
     AppleTalk ending cable range [3]: 3
     AppleTalk zone name [myzone]: ZZ Serial
     AppleTalk additional zone name:
   Configure IPX on this interface? [no]: yes
     IPX network number [2]: 3

Configuring interface Serial1:

   Is this interface in use? [yes]:
   Configure IP on this interface? [yes]:
   Configure IP unnumbered on this interface? [yes]:
     Assign to which interface [Ethernet0]:
   Configure AppleTalk on this interface? [yes]:
     Extended AppleTalk network? [yes]:
```

```
     AppleTalk starting cable range [2]:
     AppleTalk ending cable range [2]:

     AppleTalk zone name [ZZ Serial]:
      AppleTalk additional zone name:
   Configure IPX on this interface? [yes]:
   IPX network number [2]:
Configuring interface Async1:
   IPX network number [4]:
   Default client IP address for this interface [none]: 172.16.72.4
Configuring interface Async2:
   IPX network number [5]:
   Default client IP address for this interface [172.16.72.5]:
Configuring interface Async3:
   IPX network number [6]:
   Default client IP address for this interface [172.16.72.6]:
Configuring interface Async4:
   IPX network number [7]:
   Default client IP address for this interface [172.16.72.7]:
Configuring interface Async5:
   IPX network number [8]:
   Default client IP address for this interface [172.16.72.8]:
Configuring interface Async6:
   IPX network number [9]:
   Default client IP address for this interface [172.16.72.9]:
Configuring interface Async7:
   IPX network number [A]:
   Default client IP address for this interface [172.16.72.10]:
Configuring interface Async8:
   IPX network number [B]:
   Default client IP address for this interface [172.16.72.11]:
Configuring interface Async9:
   IPX network number [C]:
   Default client IP address for this interface [172.16.72.12]:
Configuring interface Async10:
   IPX network number [D]:
   Default client IP address for this interface [172.16.72.13]:
Configuring interface Async11:
   IPX network number [E]:
   Default client IP address for this interface [172.16.72.14]:
Configuring interface Async12:
   IPX network number [F]:
   Default client IP address for this interface [172.16.72.15]:
Configuring interface Async13:
   IPX network number [10]:
   Default client IP address for this interface [172.16.72.16]:
Configuring interface Async14:
   IPX network number [11]:
   Default client IP address for this interface [172.16.72.17]:
Configuring interface Async15:
   IPX network number [12]:
   Default client IP address for this interface [172.16.72.18]:
Configuring interface Async16:
   IPX network number [13]:
```

```
   Default client IP address for this interface [172.16.72.19]:

The following configuration command script was created:

hostname Router
enable secret 5 $1$krIg$emfYm/1OwHVspDuS8Gy0K
enable password ww
line vty 0 4
password ww
snmp-server community public
!
no decnet routing
appletalk routing
ipx routing

ip routing
!
line 1 16
speed 57600
flowcontrol hardware
modem inout
!
arap network 20 ARA Dialins
line 1 16
arap enable
autoselect
!
! Turn off IPX to prevent network conflicts.
interface Ethernet0
no ipx network
interface Serial0
no ipx network
interface Serial1
no ipx network
!
interface Ethernet0
ip address 172.16.72.2 255.255.255.0
appletalk cable-range 1-1 1.204
appletalk zone Sales
ipx network 1
no mop enabled
!
interface Serial0
no shutdown
no ip address
ip unnumbered Ethernet0
appletalk cable-range 3-3
appletalk zone ZZ Serial
ipx network 3
no mop enabled
!
interface Serial1
no ip address
ip unnumbered Ethernet0
```

```
appletalk cable-range 2-2 2.2
appletalk zone ZZ Serial
ipx network 2
no mop enabled
!
Interface Async1
ipx network 4
ip unnumbered Ethernet0
async default ip address 172.16.72.4
async mode interactive
!
Interface Async2
ipx network 5
ip unnumbered Ethernet0
async default ip address 172.16.72.5
async mode interactive
!
Interface Async3
ipx network 6
ip unnumbered Ethernet0
asyncdefault ip address 172.16.72.6
async mode interactive
!
Interface Async4
ipx network 7
ip unnumbered Ethernet0

async default ip address 172.16.72.7
async mode interactive
async dynamic address
!
Interface Async5
ipx network 8
ip unnumbered Ethernet0
async default ip address 172.16.72.8
async mode interactive
!
Interface Async6
ipx network 9
ip unnumbered Ethernet0
async default ip address 172.16.72.9
async mode interactive
!
Interface Async7
ipx network A
ip unnumbered Ethernet0
async default ip address 172.16.72.10
asyncmode interactive
!
Interface Async8
ipx network B
ip unnumbered Ethernet0
async default ip address 172.16.72.11
async mode interactive
```

```
!
Interface Async9
ipx network C
ip unnumbered Ethernet0
async default ip address 172.16.72.12
async mode interactive
!
Interface Async10
ipx network D
ip unnumbered Ethernet0
async default ip address 172.16.72.13
async mode interactive
!
Interface Async11
ipx network E
ip unnumbered Ethernet0
async default ip address 172.16.72.14
async mode interactive
!
Interface Async12
ipx network F
ip unnumbered Ethernet0
async default ip address 172.16.72.15
async mode interactive
!
Interface Async13
ipx network 10
ip unnumbered Ethernet0
async default ip address 172.16.72.16
async mode interactive
!
Interface Async14
ipx network 11
ip unnumbered Ethernet0
async default ip address 172.16.72.17
async mode interactive
!
Interface Async15

ipx network 12
ip unnumbered Ethernet0
async default ip address 172.16.72.18
async mode interactive
!
Interface Async16
ipx network 13
ip unnumbered Ethernet0
async default ip address 172.16.72.19
async mode interactive
!
router igrp 15
network 172.16.0.0
!
end
```

```
Use this configuration? [yes/no]: yes

Building configuration...

Use the enabled mode 'configure' command to modify this configuration.

Router#
```

Use the Streamlined Setup Facility

The streamlined setup command facility is available only if you router is running from ROM monitor and has RXBOOT ROMs installed. The streamlined setup command facility permits your router to load a system image from a network server when there are problems with the startup configuration. The Cisco IOS software automatically puts you in the streamlined setup command facility when your router is accidentally or intentionally rebooted (or you are attempting to load a system image from a network server) after:

- You issued an **`erase startup-config`** command, thereby deleting the startup configuration file.
- You have bit 6 (ignore NVRAM configuration) set in the configuration register.
- Your startup configuration has been corrupted.
- You configured the router to boot from a network server (the last four bits of the configuration register are not equal to 0 or 1) and there is no Flash or no valid image in Flash.
- You configured the router to boot the RXBOOT image.

The streamlined setup command facility differs from the standard setup command facility because the streamlined facility does not ask you to configure global router parameters. You are prompted only to configure interface parameters, which permit your router to boot.

This example shows a router entering the streamlined setup command facility:

```
         — System Configuration Dialog —
Default settings are in square brackets '[]'.
Configuring interface IP parameters for netbooting:
```

Note: ***The message Configuring interface IP parameters for netbooting only appears if you are booting over a network server and your configuration has insufficient IP information.***

The streamlined setup command facility continues, prompting you fo interface parameters for each installed interface. The facility asks if an interface is in use. If so, the facility prompts you to provide an IP ad dress and subnet mask bits for the interface. Enter the subnet mask bits as a decimal value, such as 5.

This example shows the portion of the streamlined setup command facility that prompts for interface parameters. In the example, the facility is prompting for Ethernet0 interface parameters and Serial0 interface parameters:

```
Configuring interface Ethernet0:
   Is this interface in use? [yes]:
   Configure IP on this interface? [yes]:
     IP address for this interface: 192.195.78.50
     Number of bits in subnet field [0]: 5
     Class C network is 192.195.78.0, 5 subnet bits; mask is 255.255.255.248
   Configuring interface Serial0: Is this interface in use? [yes]:
   Configure IP on this interface? [yes]:
     IP address for this interface: 192.195.78.34
     Number of bits in subnet field [5]:
     Class C network is 192.195.78.0, 5 subnet bits; mask is 255.255.255.248
```

The configuration information you provide on this screen is temporary and exists only so you can boot your system. When you reload the system, your original configuration remains intact. If your startup configuration is corrupted, enter the setup command facility and configur the basic parameters. Then issue the **`copy running-config startup-config`** command to write this configuration to NVRAM.

Chapter 6: Using the System Configuration Dialog

Introduction

This chapter describes the System Configuration Dialog process using a sample configuration. The System Configuration Dialog can be manually used to configure the router instead of using AutoInstall.

System Configuration Dialog

If you do not plan to use AutoInstall (refer to the Using AutoInstall chapter), make sure all the WAN cables are disconnected from the router. This will prevent the router from attempting to the run the AutoInstall process. The router will attempt to run AutoInstall whenever you power it on if there is a WAN connection on both ends and the router does not have a configuration file stored in NVRAM. It can take several minutes for the router to determine that AutoInstall is not set up to a remote TCP/IP host.

If your router does not have a configuration (setup) file and you are not using AutoInstall, the router will automatically start the setup com mand facility. An interactive dialog called the System Configuration Dialog appears on the console screen. This dialog helps you navigat through the configuration process by prompting you for the configuration information necessary for the router to operate.

Many prompts in the System Configuration Dialog include default responses, which are included in square brackets following the question. To accept a default answer, press <Enter>. Otherwise, type your response.

This section gives an example configuration using the System Configuration Dialog. When you are configuring your router, respond as ap propriate for your network.

At any time during the System Configuration Dialog, you can request help by typing a question mark (?) at a prompt.

Before proceeding with the System Configuration Dialog, obtain from your system administrator the node addresses and the number of bits in the subnet field (if applicable) of the Ethernet and synchronous serial ports.

Take these steps to configure the router using the System Configura tion Dialog:

1. Connect a console terminal to the console connector on theAI2524.

Note: ***The default parameters for the console port are 9600 baud, 8 data bits, no parity, and 2 stop bits.***

2. After about 30 seconds, information similar to the following is displayed on the console screen.

Note: ***The messages displayed vary, depending on the Cisco IOS feature set you selected. The screen displays in this section are for reference only and may not exactly reflect the screen displays on your console.***

When you see this information, you have successfully booted your router:

```
System Bootstrap, Version X.X(XXXX) [XXXXX XX], RELEASE SOFTWARE
Copyright (c) 1986-1992 by Cisco Systems 2500 processor with 4096 Kbytes of main
memory
Notice: NVRAM invalid, possibly due to write erase.
F3: 5797928+162396+258800 at 0x3000060

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 Right 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
Cisco Internetwork Operating System Software
IOS (tm) X000 Software (IGS-J-L), Version XX.X(XXXX) [XXXXX XXX]
Copyright (c) 1986-1996 by Cisco Systems, Inc.
Compiled Fri 20-Oct-95 16:02 by XXXXX
Image text-base: 0x03030FC0, data-base: 0x00001000
Cisco 252X (68030) processor (revision A) with 4092K/2048K bytes of memory.
Processor board ID 00000000
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version X.X, NET2, BFE and GOSIP compliant.
TN3270 Emulation software (copyright 1994 by TGV Inc).
Basic Rate ISDN software, Version X.X.
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
1 ISDN Basic Rate interface.
```

```
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read ONLY)
Notice: NVRAM invalid, possibly due to write erase.
--- System Configuration Dialog ---
At any point you may enter a question mark '?' for help. Refer to the 'Getting
Started' Guide for additional help. Use ctrl-c to abort configuration dialog at
any prompt. Default settings are in square brackets '[]'.
Would you like to enter the initial configuration dialog? [yes]:
```

3. Press <Enter> or type yes to begin the configuration process.
4. When the System Configuration Dialog asks whether you want to view the current interface summary, press <Enter> or type yes.

```
First, would you like to see the current interface summary? [yes]:
Any interface listed with OK? value "NO" does not have a valid configuration

Interface        IP-Address  OK?      Method    Status    Protocol
Ethernet0        unassigned  NO       not set   up        down
BRI0             unassigned  NO       not set   up        up
Serial0          unassigned  NO       not set   down      down
Serial1          unassigned  NO       not set   down      down
```

5. Configure the global parameters. Choose which protocols to support on the Ethernet interface. For IP installations, you can press <Enter> to accept the default values (in brackets) for most of the questions. A typical configuration is:

```
Configuring global parameters:
Enter host name [Router]:
```

Next, you are prompted to enter an enable secret password. There are two types of privileged-level passwords:

- Enable secret password (a very secure, encrypted password
- Enable password (a less secure, nonencrypted password)

The enable password is used when the enable secret password does not exist.

For maximum security, be sure the passwords are different. If you enter the same password for both, the router will accept your entry, but will display a warning message indicating that you should enter a different password.

6. Enter an enable secret password:

```
The enable secret is a one-way cryptographic secret used instead of the enable
password when it exists.
   Enter enable secret: pail
The enable password is used when there is no enable secret and when using older
software and some boot images.
```

7. Enter the enable and virtual terminal passwords:

```
Enter enable password: shovel
Enter virtual terminal password: vterm1
```

8. Press <Enter> to accept Simple Network Management Protocol (SNMP) management, or type no to refuse it:

```
Configure SNMP Network Management? [yes]: no
```

9. Configure the appropriate protocols for your router:

```
Configure Vines? [no]:
Configure LAT? [no]:
Configure AppleTalk? [no]: yes
Multizone networks? [no]: yes
Configure DECnet? [no]:
Configure IP? [yes]:
Configure IGRP routing? [yes]:
Your IGRP autonomous system number [1]: 15
Configure CLNS? [no]:
Configure bridging? [no]:
Configure IPX? [no]: yes
Configure XNS? [no]:
Configure Apollo? [no]:
```

10. Enter the ISDN BRI switch type for the router. The ISDN switch type appropriate for the router depends on the ISDN provider's equipment.

```
Enter ISDN BRI Switch Type [none]: basic-5ess
```

Refer to this table for ISDN switch types:

Country	ISDN Switch Type	Description
Australia	basic-ts013	Australian TS013 switches
Europe	basic-1tr6	German 1TR6 ISDN switches
	basic-nwnet3	Norwegian NET3 ISDN switches (phase 1)
	basic-net3	NET3 ISDN switches (UK and others)
	vn2	French VN2 ISDN switches
	vn3	French VN3 ISDN switches
Japan	ntt	Japanese NTT ISDN switches
North America	basic-5ess	AT&T basic rate switches
	basic-dms100	NT DMS-100 basic rate switches
	basic-ni1	National ISDN-1 switches
New Zealand	basic-nznet3	New Zealand NET3 switches

Chapter 7: Manually Loading System Images

Introduction

This chapter outlines the steps to load and maintain system images, microcode images, and configuration files. These instructions describe copying system images from routers to network servers (and vice versa), displaying and comparing different configuration files, and listing the Cisco IOS software version running on the router.

This chapter also explains how to manually load system images from ROM monitor so you can successfully boot the router when typical startup processes malfunction.

- System images contain the system software.
- Microcode images contain microcode to be downloaded to various hardware devices.
- Configuration files contain commands entered to customize th function of the Cisco IOS software.

To benefit most from these instructions, your router must contain minimal configuration that allows you to interact with the system software. You can create a basic configuration file using the setup com mand facility. See the user guide for your hardware platform for more information on using setup at first-time startup. Refer to Use Setup after First-Time Startup, for more information.

Image and Configuration File Load Task List

To load and maintain system images, microcode images, and configuration files needed for startup, complete the tasks outlined in the next section.

Note: ***The organization of tasks assumes you have a minimal configuration that you want to modify.***

The tasks in the first three sections are typical for all routers. Perform the remaining tasks as needed for your routing environment.

- Retrieve System Images and Configuration Files
- Perform General Startup Tasks
- Store System Images and Configuration Files
- Perform Startup Tasks
- Manually Load a System Image from ROM Monitor

Retrieve System 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 you routing environment. This section describes tasks related to retrieving system images and configuration files for modification.

Retrieve System Images and Configuration File Task List

When retrieving system images and configuration files, perform these tasks. The first two are required; the rest are optional.

- Copy System Images from a Network Server to Flash Memory
- Copy Configuration Files from a Network Server to the Router
- Change the Buffer Size for Loading Configuration Files
- Verify the Image in Flash Memory
- Display System Image and Configuration Information
- Reexecute the Configuration Commands in Startup Configuration
- Clear the Configuration Information

Copy System Images from a Network Server to Flash Memory

You can copy system images from a Trivial File Transfer Protocol (TFTP) server to Flash memory:

1. Make a backup copy of the current system software image. Refer to Copy System Images from Flash Memory to a Network Server, for more information.

2. Copy a system image to Flash memory.

   copy tftp flash

3. When prompted, enter the server IP address or domain name.

   ip-address **or** ***name***

4. If prompted, enter the server system filename.

   filename

5. If prompted, enter the Flash memory device that is to receive th copy of the system image.

   device

Note: ***Be sure there is enough available space before copying a file to Flash memory. Use the*** `show flash` ***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 will continue, but the entire file will not be copied into Flash memory. The failure message*** `buffer overflow - xxxx/ xxxx` ***will appear, where*** *`xxxx/xxxx`* ***is the number of bytes read in relation to the number of bytes available.***

When you enter 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 router serving ROM or Flash system soft ware images. The system then prompts you for the filename of the software image to copy.

For the **`copy tftp flash`** and **`copy tftp`** ***`file-id`*** commands, the router gives you the option of erasing the existing Flash memory before writing to it when there is space available to do so. If there is no free Flash memory available, or if the Flash memory has never been written to, you must run the erase routine before copying new files. The system will inform you of these conditions and prompt 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, entering only **`copy tftp`** ***`filename`***. When you omit the device, the system uses the default device specified by th **`cd`** command.

If you try to copy a file into Flash memory and that file is already in Flash memory, a prompt informs you that a file with the same nam already exists. The new file replaces the existing file. The first copy o the file still resides within Flash memory, but it is rendered unusable in favor of the newer 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 and is invalid. In this case, the original file in Flash memory is still available to the system.

This example demonstrates the use of the **`copy tftp flash`** command to copy a system image named gs7-k when Flash memory is too full to copy the file. The filename gs7-k can be in lowercase or uppercase; the system sees GS7-K as gs7-k. If more than one file of the sam name is copied to Flash, regardless of case, the last file copied is th valid file.

```
env-chassis# copy tftp flash
IP address or name of remote host [255.255.255.255]? dirt
Translating "DIRT"...domain server (255.255.255.255) [OK]

Name of file to copy? gs7-k
Copy gs7-k from 131.108.13.111 into flash memory? [confirm]
Flash is filled to capacity.

Erasure is needed before flash may be written.
Erase flash before writing? [confirm]
Erasing flash EPROMs bank 0

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 1

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 2

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 3

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Loading from 131.108.1.111:!!!!...
  [OK - 1906676/4194240 bytes
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1906676, checksum = 0x12AD
```

The exclamation point (`!`) indicates that the copy process is taking place. Each exclamation point (`!`) indicates that 10packets have been transferred successfully. A series of V characters indicates that a checksum verification of the image is occurring after the image is written to Flash memory.

Note: ***If you enter* n *after the* Erase flash before writing? *prompt, the copy process continues. If you enter* y *and confirm the erasure, the erase routine begins. Be sure you have enough Flash memory space before entering* n *at the erasure prompt.***

This example demonstrates the process of copying a system image named gs7-k into the current Flash configuration when a file named gs7-k already exists:

```
env-chassis# copy tftp flash
IP address or name of remote host [131.108.13.111]?
Name of file to copy? gs7-k
File gs7-k already exists; it will be invalidated!
Copy gs7-k from 131.108.13.111 into flash memory? [confirm]
2287500 bytes available for writing without erasure.
Erase flash before writing? [confirm]n
Loading from 131.108.1.111:!!!!...
[OK - 1906676/2287500 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1902192, checksum = 0x12AD
```

The exclamation point (!) indicates that the copy process is taking place. Each exclamation point (!) indicates that 10packets have been transferred successfully. A series of V characters indicates that a checksum verification of the image is occurring after the image is written to Flash memory.

In this example, the Flash security jumper is not installed, so you cannot write files to Flash memory. Also, be sure to set the write-protect switch on the Flash memory card to unprotected.

```
Router# copy tftp flash
Flash: embedded flash security jumper(12V) must be strapped to modify flash memory
```

Note: ***To terminate this copy process, press <Ctrl-^> (th <Ctrl>, <Shift>, and <6> keys on a standard keyboard) simultaneously. Although the process terminates, the partial file copied before the termination remains until th entire Flash memory is erased.***

You can copy normal or compressed images to Flash memory. You can produce a compressed system image on any UNIX platform using the **`compress`** command. Refer to your UNIX platform's documentation for the exact usage of the **`compress`** command.

This example shows how to copy a system image named IJ09140Z into the current Flash configuration:

```
Router# copy tftp flash
IP address or name of remote host [255.255.255.255]? server1
Name of tftp filename to copy into flash []? IJ09140Z
copy IJ09140Z from 131.131.101.101 into flash memory? [confirm]
xxxxxxxx bytes available for writing without erasure.
erase flash before writing? [confirm]
Clearing and initializing flash memory (please wait)####...
Loading from 101.2.13.110:!!!!...
[OK - 324572/524212 bytes]
Verifying checksum...
 VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV... Flash verification successful. Length =
1204637, checksum = 0x95D
```

The series of pound signs #) indicates that each Flash device is being cleared and initialized—one per device. Different platforms use different methods to indicate that Flash is clearing.

The exclamation point (`!`) indicates that the copy process is taking place. Each exclamation point `!`) indicates that ten packets have been transferred successfully.

The series of `V` characters indicates that a checksum is being calculated.

An `O` indicates an out-of-order packet.

A period (`.`) indicates a timeout. The last line in the sample configuration indicates that the copy is successful.

Copy Configuration Files from a Network Server to the Router

You can copy configuration files from a TFTP server to the router. You might use this process to restore a configuration file to the router if you have backed up the file to a server. If you replace a router and want to use the configuration file that you created for the original, you can restore that file instead of recreating it. You can also use this pro cess to copy a different configuration to the router that is stored on a network server.

You can copy a configuration file from a TFTP server to the running configuration or to the startup configuration. When you copy a configuration file to the running configuration, you copy to and run the file from RAM.

When you copy a configuration file to the startup configuration, you copy it to the nonvolatile random-access memory (NVRAM).

To copy a configuration file from a TFTP server to the router, complete these tasks from EXEC mode:

1. Copy a file from a TFTP server to the router.

   `copy tftp {running-config | startup-config}`

2. When prompted, enter the server IP address or domain name.

   {*ip-address* | *name*}

3. If prompted, enter the filename of the server system image.

   filename

Change the Buffer Size for Loading Configuration Files

The buffer that holds the configuration commands is generally the size of NVRAM. Complex configurations might need a larger configura tion file buffer size. To change the buffer size:

1. Enter configuration mode from the terminal.

   `configure terminal`

2. Change the buffer size to use for booting a host or network configuration file from a network server.

   boot buffersize *bytes*

3. To exit configuration mode, press <Ctrl-Z>.

4. Save the configuration file to your startup configuration. This step saves the configuration to NVRAM.

   `copy running-config startup-config`

In this example, the buffer size is set to 50000 bytes:

```
Router1# configure terminal
Router1(config)# boot buffersize 50000
^Z
Router1# copy running-config startup-config
```

Verify 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. The check sum of the image in Flash memory is displayed at the bottom of the screen when you issue th **`copy tftp flash`** command. The RE ADME file is copied to the network server automatically when you install the system software image on the server.

Note: ***If the checksum value does not match the value in the*** **README** ***file, do not reboot the router. Instead, issue the*** **`copy`** ***command and compare the checksums again. If the checksum is repeatedly wrong, copy the original system software image back into Flash memory before you reboot the router from Flash memory. If you have a corrupted image in Flash memory and try to boot from Flash, the router will start 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 router will not function and must be reconfigured through a direct console port connection.***

Display System Image and Configuration Information

Perform these tasks in EXEC mode to display information about system software, system image files, and configuration files:

1. List information about Flash memory, including system image filenames and amounts of used and remaining memory.

   `show flash`

2. List information about Flash memory, including system image filenames, amounts of memory used and remaining, and Flash partitions.

   ```
   show flash [all | chips | detailed | err |
   partition number [all | chips | detailed |
   err] | summary]
   ```

3. View the console output generated during the Flash load helpe operation.

   ```
   show flh-log
   ```

4. List the configuration information in running memory.

   ```
   show running-config
   ```

5. List the startup configuration information. The startup configuration is usually NVRAM.

   ```
   show startup-config
   ```

6. List the system software release version, configuration register setting, and so on.

   ```
   show version
   ```

You can also use th **o** command in ROM monitor mode to list th configuration register settings on some models.

The Flash memory content listing does not include the checksum of individual files. To recompute and verify the image checksum after th image is copied into Flash memory, type the following in EXEC mode:

```
verify flash
```

When you enter this command, the screen prompts you for the filename to verify. By default, it prompts for the last (most recent) file in Flash. Press <Enter> to recompute the default file checksum, or enter the filename of a different file at the prompt. Note that the checksum for microcode images is always 0x0000.

This example illustrates how to use this command:

```
Router# verify flash

Name of file to verify [gsxx]?
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1923712, checksum = 0xA0C1
Router#
```

Reexecute the Configuration Commands in Startup Configuration

You can reexecute the configuration commands stored in NVRAM.

To reexecute the commands located in the startup configuration, type this command in privileged EXEC mode:

```
configure memory
```

Clear the Configuration Information

To clear the contents of your startup configuration, type this command in EXEC mode:

```
erase startup-config
```

If you try 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 try to erase or delete the last valid system image specified in the BOOT environment vari able, the system prompts you to confirm this deletion.

This example erases the myconfig file from a Flash memory card inserted in slot 0:

```
Router# erase slot0:myconfig
```

This example deletes the myconfig file from a Flash memory card inserted in slot 0:

```
Router# delete slot0:myconfig
```

Perform General Startup Tasks

When modifying your routing environment, you perform 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 router 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 that the router uses to start up.

General Startup Task List

General startup tasks include:

- Enter Configuration Mode and Select a Configuration Source
- Modify the Configuration Register Boot Field
- Specify the Startup Configuration Fil

Enter Configuration Mode and Select a Configuration Source

To enter configuration mode, enter th **`configure`** command at the privileged EXEC prompt. The Cisco IOS software responds with this prompt by asking you to specify the terminal or memory or a file stored on a network server (network) as the source of configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

These methods are described in these sections:

- Configure the Cisco IOS software from the Terminal
- Configure the Cisco IOS software from Memory
- Configure the Cisco IOS software from the Network

The Cisco IOS software accepts one configuration command per line. You can enter as many configuration commands as you want.

You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are not stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with th **`show running-config`** EXEC command. Also, when the startup configuration is NVRAM, comments do not show up when you list the startup configuration with th **`show startup-config`** EXEC command. Comments are stripped out of the configuration file when it is loaded onto the router. However, you can list the comments in configuration files stored on a TFTP server.

Configure the Cisco IOS software from the Terminal

When you configure the software from the terminal, it executes the commands you enter at the system prompts. To configure the software from the terminal:

1. Enter configuration mode and select the terminal option.

   `configure terminal`

2. Enter the necessary configuration commands.

3. To exit configuration mode, press <Ctrl-Z>.

4. Save the configuration file to your startup configuration. This step saves the configuration to NVRAM.

   `copy running-config startup-config`

In this example, the software is configured from the terminal. The comment `The following command provides the router host name` identifies the purpose of the next command line. The **`hostname`** command changes the router name from router1 to router2. By pressing <Ctrl-Z>, the user quits configuration mode. Finally, the **`copy running-config startup-config`** command saves the current configuration to the startup configuration.

```
Router1# configure terminal
Router1(config)#
! The following command provides the router host name.
Router1(config)# hostname router2
^Z
Router2# copy running-config startup-config
```

When the startup configuration is NVRAM, it stores the current configuration information in text format as configuration commands, recording only nondefault settings. The memory is checksummed to guard against corrupted data.

The startup software always checks for configuration information in NVRAM. If NVRAM holds valid configuration commands, the Cisco IOS software executes the commands automatically at startup. If th software detects a problem with NVRAM or the configuration it contains, it enters setup mode and prompts for configuration. Problems can include a bad checksum for the information in NVRAM or the absence of critical configuration information. Refer to Use Setup for Configuration Changes, for details on the setup command facility.

Configure the Cisco IOS software from Memory

To configure the software to execute the commands located in NVRAM or to execute the configuration specified by the CONFIG_FILE environment variable, type this command in privileged EXEC mode:

```
configure memory
```

Configure the Cisco IOS software from the Network

You can configure the software by retrieving and modifying a configuration file stored on one of your network servers:

1. Enter configuration mode with the network option.

   ```
   copy tftp running-config
   ```

2. 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 one network server in particular.

   ```
   {host | network}
   ```

3. At the system prompt, enter the optional IP address of the remote host from which you are retrieving the configuration file.

   ```
   ip-address
   ```

4. At the system prompt, enter the name of the configuration file or accept the default name.

   ```
   filename
   ```

5. Confirm the configuration filename that the system supplies.

   ```
   y
   ```

In this example, the software is configured from the file tokyo-config at IP address 131.108.2.155:

```
Router1# copy tftp running-config
Host or network configuration file [host]?
IP address of remote host [255.255.255.255]? 131.108.2.155
Name of configuration file [tokyo-confg]?
Configure using tokyo-confg from 131.108.2.155? [confirm] y
Booting tokyo-confg from 131.108.2.155:!!!
[OK - 874/16000 bytes]
```

Copy 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 task loads a configuration file directly into NVRAM in a location specified by th CONFIG_FILE environment variable.

To copy a configuration file directly to the startup configuration, type this command in EXEC mode:

```
copy tftp startup-config
```

Modify the Configuration Register Boot Field

The configuration register boot field determines whether the router loads an operating system image and, if so, where it obtains this system image. The next sections describe the process for using the configuration register boot field, your process for setting this field, and the tasks you must perform to modify the configuration register boot field.

How the Router Uses 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 whether the router loads an operating system and where it obtains the system image:

- When the entire boot field equals 0-0-0-0, the router does not load a system image. Instead, it 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 0-0-0-1, the router loads the system image found in boot ROMs.

- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the router 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, th router loads a default system image stored on a network server.

When loading a default system image from a network server, the router uses the configuration register settings to determine the default system image filename for booting from a network server. The router forms the default boot filename by starting with the word cisco and then appending the octal equivalent of the boot field number in th configuration register, followed by a hyphen (-) and the processor type name (cisconn-cpu). See the appropriate hardware installation guide for details on the configuration register and default filename.

Setting the Boot Field

You must correctly set the configuration register boot field to ensure that your router loads the operating system image as you intend. To set the boot field:

1. Obtain the current configuration register setting. This setting is hexadecimal value.

2. Modify the current configuration register setting to reflect the way you want to load a system image. To do so, change the least significant hexadecimal digit to:

   0 to load the system image manually using the **`boot`** command in ROM monitor mode.

   1 to load the system image from boot ROMs.

   2 through F to load 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, you would change the configuration register setting to 0x102.

1. Reboot the router to make your changes to the configuration register take effect.

Perform the Boot Field Modification Tasks

You modify the boot field from the hardware configuration register or the software configuration register, depending on the platform.

The hardware configuration register can be changed only on the processor card or with dual in-line package (DIP) switches located at the back of the router. For information on modifying the hardware configuration register, refer to the appropriate hardware installation guide.

To modify the software configuration register boot field, complet these tasks:

1. Obtain the current configuration register setting.

   `show version`

2. Enter configuration mode, selecting the terminal option.

   `configure terminal`

3. Modify the existing configuration register setting to reflect the way in which you want to load a system image.

   `config-register` ***`value`***

4. To exit configuration mode, press <Ctrl-Z>.

5. Reboot the router to make your changes take effect.

   `reload`

Use the **`show version`** EXEC command to display the current configuration register setting. In ROM monitor mode, use the **`o`** command to list the value of the configuration register boot field.

In this example, the **`show version`** command indicates the current configuration register is set so the router 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 in structs the router to a load a system image from commands in the startup configuration file or from a default system image stored on a network server.

```
Router1# show version
GS Software, Version 9.0(1)
Copyright (c) 1986-1992 by Cisco Systems, Inc.
Compiled Fri 14-Feb-92 12:37
System Bootstrap, Version 4.3
Router1 uptime is 2 days, 10 hours, 0 minute
System restarted by reload
System image file is unknown, booted via tftp from 131.108.13.111
Host configuration file is "thor-boots", booted via tftp from 131.108.13.111
Network configuration file is "network-confg", booted via tftp from 131.108.13.111

CSC3 (68020) processor with 4096K bytes of memory.
X.25 software.
Bridging software.
1 MCI controller (2 Ethernet, 2 Serial).
2 Ethernet/IEEE 802.3 interface.
2 Serial network interface.
32K bytes of non-volatile configuration memory.
Configuration register is 0x0

Router1# configure terminal
Router1(config)# config-register 0xF
^Z
Router1# reload
```

Specify the Startup Configuration File

Configuration files can be stored on network servers. You can configure the router to automatically request and receive two configuration files from the network server at startup:

- Network configuration fil
- Host configuration file

The server first attempts to load the network configuration file. This file contains information shared among several routers. For example, you can use it to provide mapping between IP addresses and host names.

The second file the server attempts to load is the host configuration file, containing commands applicable to one router in particular. Both the network and host configuration files must reside on a network server reachable via TFTP and must be readable.

You can specify an ordered list of network configuration and host configuration filenames. The Cisco IOS software scans this list until it successfully loads the appropriate network or host configuration file.

Specify the Startup Configuration File Task List

To specify a startup configuration file, perform either the first two tasks or the third task:

- Download the Network Configuration File
- Download the Host Configuration Fil

Download the Network Configuration File

To configure the Cisco IOS software to download a network configu ration file from a server at startup:

1. Enter configuration mode from the terminal.

   `configure terminal`

2. Enter the network configuration filename to download a file using TFTP.

   `boot network tftp` ***`filename`*** **`[`*`ip-address`*`]}`**

3. Enable the router to automatically load the network file upon re start.

   `service config`

4. To exit configuration mode, press <Ctrl-Z>.

1. Save the configuration file to your startup configuration. On most platforms, this step saves the configuration to NVRAM.

   `copy running-config startup-config`

For Step 2, if you do not specify a network configuration filename, th Cisco IOS software uses the default filename network-confg. If you omit the tftp keyword, the software assumes that you are using TFTP to transfer the file and that the server whose IP address you specify supports TFTP.

You can specify more than one network configuration file. The software tries them in order until it loads one successfully. This procedur can be useful for keeping files with different configuration information loaded on a network server.

Download the Host Configuration File

To configure the Cisco IOS software to download a host configuration file from a server at startup, complete the following tasks. Step 2 is optional. If you do not specify a host configuration filename, the router uses its own name to form a host configuration filename by converting the name to all lowercase letters, removing all domain information, and appending -confg. If no host name information is available, the software uses the default host configuration filename router-confg.

1. Enter configuration mode from the terminal.

   ```
   configure terminal
   ```

2. Optionally, enter the host configuration filename to be downloaded.

   ```
   boot host {tftp filename [ip-address]}
   ```

3. Enable the device to automatically load the host file upon restart.

   ```
   service config
   ```

4. To exit configuration mode, press <Ctrl-Z>.

5. Save the configuration file to your startup configuration. This step saves the configuration to NVRAM.

   ```
   copy running-config startup-config
   ```

6. Reset the router with the new configuration information.

   ```
   reload
   ```

You can specify more than one host configuration file. The Cisco IOS software tries them in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

In this example, a router is configured to boot from the host configuration file hostfile1 and from the network configuration file networkfile1:

```
Router1# configure terminal
Router1(config)# boot host hostfile1
Router1(config)# boot network networkfile1
Router1(config)# service config
^Z
Router1# copy running-config startup-config
```

If the network server fails to load a configuration file during startup, it tries again every 1 0minutes (the default setting) 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 un able to load the specified file, it displays this message:

```
Booting host-confg... [timed out]
```

If there are any problems with the startup configuration file, or if th configuration register is set to ignore NVRAM, the router enters the setup command facility. Refer to Use Setup for Configuration Change , for details on the setup command.

Store System Images and Configuration Files

After modifying and saving your routing environment's unique configurations, you might want to store them on a network server. You can use these network server copies of system images and configuration files as backup copies.

Store System Images and Configuration Files Task List

To store system images and configuration files:

- Copy System Images from Flash Memory to a Network Server
- Copy Configuration Files from the Router to a Network Server

Copy System Images from Flash Memory to a Network Server

You can copy system images from Flash memory to a TFTP 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.

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 EXEC mode:

1. (Optional) If you do not already know it, learn the exact spelling of the system image filename in Flash memory.

   ```
   show flash all
   ```

2. Copy the system image from Flash memory to a TFTP server.

   copy flash tftp

3. When prompted, enter the IP address or domain name of the TFTP server.

   {*ip-address* | *name*}

4. When prompted, enter the filename of the system image in Flash memory.

   filename

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 (xk09140z) is listed near the end of the **show flash all** output.

```
Router# show flash all
2048K bytes of flash memory on embedded flash (in XX).

ROM       socket        code          bytes         name
0         U42           89BD          0x40000       INTEL 28F020
1         U44           89BD          0x40000       INTEL 28F020
2         U46           89BD          0x40000       INTEL 28F020
3         U48           89BD          0x40000       INTEL 28F020
4         U41           89BD          0x40000       INTEL 28F020
5         U43           89BD          0x40000       INTEL 28F020
6         U45           89BD          0x40000       INTEL 28F020
7         U47           89BD          0x40000       INTEL 28F020

   security jumper(12V) is installed,
   flash memory is programmable.
file      offset        length        name
0         0x40          1204637       xk09140z
   [903848/2097152 bytes free]

Router# copy flash tftp
IP address of remote host [255.255.255.255]? 101.2.13.110
filename to write on tftp host? xk09140z
writing xk09140z !!!!...
successful tftp write.
Router#
```

The exclamation point (!) indicates that the copy process is taking place. Each exclamation point !) indicates that ten packets have been transferred successfully. To stop the copy process, press <Ctrl-^>.

Once you have configured Flash memory, you might want to configure the system (using th **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.

Copy Configuration Files from the Router to a Network Server

You can copy configuration files from the router to a TFTP server. You might do this to back up a current configuration file to a server before changing its contents, allowing you to later restore the original.

Usually, the configuration file that you copy to must already exist on the TFTP server and be globally writable before the TFTP server al lows you to write to it.

To store configuration information on a TFTP network server, complete these tasks in the EXEC mode:

1. Specify that the running or startup configuration file be stored on a network server.

   copy {running-config | startup-config} tftp

2. Enter the IP address of the network server.

   ip-address

3. Enter the name of the configuration file to store on the server.

   filename

4. Confirm the entry.

   y

The command prompts you for the destination host's address and a filename, as the following example illustrates. This example copies configuration file from a router to a TFTP server:

```
Tokyo# copy running-config tftp
Remote host [131.108.2.155]?
Name of configuration file to write [tokyo-confg]?
Write file tokyo-confg on host 131.108.2.155?[confirm] y
#
Writing tokyo-confg!!! [OK]
```

Perform Startup Tasks

The startup tasks in this section are optional.

Startup Task List

You can perform these optional startup tasks:

- Partition Flash Memory Using Dual Flash Bank
- Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems

Partition Flash Memory Using Dual Flash Bank

Dual Flash bank allows you to partition banks of Flash memory into separate, logical devices so that the router can hold and maintain two different software images. (A bank is a set of four chips.) No downtime is required: you can write software into Flash memory while running software in another bank of Flash memory.

Systems that Support Dual Flash Bank

To use dual Flash bank, you must have at least two banks of Flash memory. The minimum partition size is the size of a bank.

Dual Flash bank is supported on low-end systems that have at least two banks of Flash memory, including systems that support a single SIMM that has two banks of Flash memory.

Benefits

Partitioning Flash memory provides these benefits:

- For any system, partitioning—rather than having one logical Flash memory device—provides a cleaner way of managing different files in Flash memory, especially if the Flash memory size is large.
- For systems that execute code out of Flash memory, partitioning allows you to download a new image into the file system in one Flash memory bank while an image is being executed from the file system in the other bank. The download is simple and causes no network disruption or downtime. After the download is complete, you can switch over to the new image at a convenient time.
- One system can hold two different images, one image acting as backup for the other. Therefore, if a downloaded image fails to boot for some reason, the earlier running, good image is still available. Each bank is treated as a separate device.

Flash Load Helper versus Dual Flash Bank

You might use Flash load helper rather than dual Flash bank for one of these reasons:

- You want to download a new file into the same bank from which the current system image is executing.
- You want to download a file that is larger than the size of a bank, and hence want to switch to a single-bank mode.
- You have only one single-bank Flash SIMM installed. In this case, Flash load helper is the best option for upgrading your software.

Refer to Use Flash Load Helper to Upgrade Software on Run-from Flash Systems, for more information about working with Flash load helper.

Understanding Relocatable Images

Because partitioning requires that run-from-Flash images be loaded into different Flash memory banks at different physical addresses, images must be relocatable. A relocatable image is an image that contains special relocation information that allows:

- The image to relocate itself whenever it is loaded into RAM fo execution
- A download program with appropriate support to relocate the im age before it is stored in Flash memory so the image can run in place in Flash memory, regardless of where in Flash memory it is stored

Run-from-Flash systems formerly ran nonrelocatable images that needed to be stored in Flash memory at a specific address. As a result, the image had to be stored as the first file in Flash memory. If th image was stored at any other location in Flash memory, it could not be executed in Flash memory, nor could the image be executed from RAM. The relocatable image overcomes this limitation.

With Flash partitioning, the run-from-Flash images will not work unless they are loaded into the first device as the first file. This require ment defeats the purpose of partitioning. However, relocatable images can be loaded into any Flash partition (and not necessarily as the first file within the partition) and executed in place.

Unless the image is downloaded as the first file in the first partition, this download must be performed by an image that recognizes relocatable images.

In contrast, a nonrelocatable image is an image that does not recognize relocatable images.

You can identify a relocatable image by its name. The naming convention for images that are stored on a UNIX system is:

platform-capabilities-type

The letter l in the type field indicates a relocatable image. Examples of some relocatable image names include

- **igs-i-l**—IP-only imag
- **igs-d-l**—Desktop feature image
- **igs-bpx-l**—Enterprise image

Only images with the igs prefix used by the AI2524 are available as relocatable images. Images distributed on floppy diskettes might have different naming conventions.

For backward compatibility, the relocatable images are linked to execute as the first file in the first Flash memory bank. This makes the images similar to previous Flash memory images. Thus, if you download a relocatable image into a nonrelocatable image system, the image runs correctly from Flash memory.

Dual Flash Bank Configuration Task List

To use dual Flash memory bank, perform these tasks:

- Partition Flash Memory
- Copy a File into a Flash Partition
- Manually Boot from Flash Memory
- Configure the Router to Automatically Boot from Flash Memory
- Configure a Flash Partition as a TFTP Serve

Refer to Display System Image and Configuration Information, for information about monitoring dual Flash bank.

To upgrade your software, you must erase Flash memory when prompted during the download. This ensures that the image is downloaded as the first file in Flash memory.

Partition Flash Memory

To partition Flash memory, type the following in global configuration mode:

```
partition flash partitions [size1 size2]
```

This task succeeds only if the system has at least two banks of Flash and if the partitioning does not cause an existing file in Flash memory to be split across the two partitions.

Copy a File into a Flash Partition

In EXEC mode, download a file into a Flash partition:

```
copy tftp flash
```

The prompts displayed after you execute these tasks indicate the method by which the file can be downloaded into each partition. The possible methods are:

None	No known way to copy into the partition.
RXBOOT-Manual	You must manually reload to the rxboot image in ROM to copy the image.
RXBOOT-FLH	The copy is automatic via the Flash load helper software in boot ROMs.
Direct	The copy is created directly.

If the image can be downloaded into more than one partition, you ar prompted for the partition number. Enter any of these commands at th partition number prompt to obtain help:

?	Display the directory listings of all partitions.
?1	Display the directory of the first partition.
?2	Display the directory of the second partition.
q	Quit the copy command.

Manually Boot from Flash Memory

To manually boot the router from Flash memory, perform one of thes tasks in ROM monitor mode:

- Boot the first bootable file found in any partition.

  boot {flash | flash flash:}

- Boot the first bootable file from the specified partition.

  boot {flash | flash flash:} ***partition-number:***

- Boot the specified file from the first partition.

  boot {flash | flash flash:} ***filename***

- Boot the specified file from the specified partition.

  boot {flash | flash flash:} ***partition-number:filename***

The result of booting a relocatable image from Flash memory depends on where and how the image was downloaded into Flash memory. This table 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 will execute in place from Flash memory, like a run-from-Flash image.
The image was downloaded not 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 relocatabl image.	The image will execute in place from Flash memory, like a run-from-Flash image.
The image was downloaded not as the first file by a relocat able image (including download into the second partition).	The relocatable image relocates the image before storage in Flash memory. Hence, the image will execute in place from Flash memory, like any other run from-Flash image.

Configure the Router to Automatically Boot from Flash Memory

To configure the router to boot automatically from Flash memory, perform one of these tasks in global configuration mode:

- Boot the first bootable file found in any partition.

  boot system {flash | flash flash:}

- Boot the first bootable file from the specified partition.

  boot system {flash | flash flash:}
  partition-number:

- Boot the specified file from the first partition.

  boot system {flash | flash flash:} ***filename***

- Boot the specified file from the specified partition.

  boot system {flash | flash flash:}
  partition-number:filename

The result of booting a relocatable image from Flash memory depends on where and how the image was downloaded into Flash memory.

Configure a Flash Partition as a TFTP Serve

To configure a Flash partition as a TFTP server, perform one of these tasks in global configuration mode:

- Specify a file.

  tftp-server flash ***filename***

- Specify a file in the first partition of Flash.

  tftp-server flash ***filename***

- Specify a file in the specified partition of Flash.

  tftp-server flash ***partition-number:***
  filename

Once you have specified TFTP server operation, exit configuration mode and save the configuration information to your startup configuration.

Use Flash Load Helper to Upgrade Software on Run-from-Flash Systems

Flash load helper is a software option that enables you to upgrade system software on run-from-Flash systems that have a single bank of

Flash memory. It is a lower-cost software upgrade solution than dual-bank Flash, which requires two banks of Flash memory on one SIMM.

The Flash load helper software upgrade process is simple and does not require additional hardware; however, it does require some brief network downtime. A system image running from Flash can use Flash load helper only if the boot ROMs support Flash load helper. Otherwise, you must perform the Flash upgrade manually. Refer to Manu ally Boot from Flash, for more information.

Flash load helper is an automated procedure that reloads the ROM-based image, downloads the software to Flash memory, and reboots to the system image in Flash memory. Flash load helper performs checks and validations to maximize the success of a Flash upgrade and mini mize the chance of leaving Flash memory either in an erased state or with a file that cannot boot.

In run-from-Flash systems, the software image is stored in and executed from the Flash EPROM rather than from RAM. This method reduces memory cost. A run-from-Flash system requires enough Flash EPROM to hold the image and enough main system RAM to hold the routing tables and data structures. The system does not need the same amount of main system RAM as a run-from-RAM system because the full image does not reside in RAM. The AI2524 is a Run-from-Flash system.

Flash load helper:

- Confirms access to the specified source file on the specified server before erasing Flash memory and reloading to the ROM image for the actual upgrade.
- Warns you if the image being downloaded is not appropriate fo the system.
- Prevents reloads to the ROM image for a Flash upgrade if the system is not set up for automatic booting and if the user is not on the console terminal. In the event of a catastrophic failure during the upgrade, Flash load helper can bring up the boot ROM image as last resort rather than forcing the system to wait at the ROM monitor prompt for input from the console terminal.
- Retries Flash downloads automatically up to six times. The retry sequence is:

  First try
  Immediate retry
  Retry after 30 seconds
  Reload ROM image and retry
  Immediate retry
  Retry after 30 seconds

- Allows you to save any configuration changes made before you exit out of the system image.
- Notifies users logged in to the system of the impending switch to the boot ROM image so that they do not lose their connections unexpectedly.
- Logs console output during the Flash load helper operation into a buffer that is preserved through system reloads. You can retrieve the buffer contents from a running image. The output is useful when console access is unavailable or when a failure occurs in the download operation.

Flash load helper can also be used on systems with multiple banks of Flash memory that support Flash memory partitioning. Flash load helper enables you to download a new file into the same partition from which the system is executing an image.

For information about how to partition multiple banks of Flash memory so your system can hold two different images, refer to Partition Flash Memory Using Dual Flash Bank.

Flash Load Helper Configuration Task List

Perform these tasks to use and monitor Flash load helper:

- Download a File Using Flash Load Helper
- Monitor Flash Load Helper

Download a File Using Flash Load Helper

To download a new file to Flash memory using Flash load helper, check to make sure that your boot ROMs support Flash load helper and then type the following in privileged EXEC mode:

```
copy tftp flash
```

This error message appears if you are in a Telnet session and the system is set for manual booting (the boot bits in the configuration register are zero):

```
ERR: Config register boot bits set for manual booting
```

In case of catastrophic failure in the Flash memory upgrade, this erro message helps to minimize the chance of the system going down to ROM monitor mode and being taken out of the remote Telnet user's control.

The system tries to bring up at least the boot ROM image if it cannot boot an image from Flash memory. Before reinitiating the **copy tftp flash** command, you must set the configuration register boot field to a nonzero value, using th **config-register** global configuration command.

The **copy tftp flash** command initiates a series of prompts to which you must provide responses. This example illustrates this dia log:

```
Router# copy tftp flash
*********************** NOTICE **************************
Flash load helper v1.0
This process will accept the TFTP copy options and then terminate the current
system image to use the ROM based image for the copy. Router functionality will
not be available during that time. If you are logged in via telnet, this
connection will terminate. Users with console access can see the results of the
copy operation. **********************************************************
```

If terminals other than the one on which this command is executed are active, this message appears:

```
There are active users logged into the system.

Proceed? [confirm] y
System flash directory:
File             Length        Name/status
1                2251320       abc/igs-kf.914
[2251384 bytes used, 1942920 available, 4194304 total]
```

Enter the IP address or the name of the remote host you are copying from:

```
Address or name of remote host [255.255.255.255]? 131.108.1.111
```

Enter the name of the file you want to copy:

```
Source file name? abc/igs-kf.914
```

Enter the name of the destination file:

```
Destination file name [default = source name]?
Accessing file `abc/igs-kf.914' on 131.108.1.111....
Loading from 131.108.13.111:
Erase flash device before writing? [confirm]
```

If you choose to erase Flash memory, the dialog continues. The **`copy tftp flash`** operation verifies the request from the running image by trying to copy a single block from the remote TFTP server. Then the Flash load helper is executed, causing the system to reload to th ROM-based system image.

```
Erase flash device before writing? [confirm] y
Flash contains files. Are you sure? [confirm] y
```

If the file is not a valid image for the system, a warning appears and the system requests a separate confirmation:

```
Copy `abc/igs-kf.914' from TFTP server
as `abc/igs-kf.914' into Flash WITH erase? y

%SYS-5-RELOAD: Reload requested
%FLH: rxboot/igs-kf.914r from 131.108.1.111 to flash...
```

If you do not erase Flash memory and there is no file duplication, th dialog continues:

```
Erase flash device before writing? [confirm] n
Copy `abc/igs-kf.914' from TFTP server
as `abc/igs-kf.914' into Flash WITHOUT erase? y
```

If you do not erase Flash memory and if there was file duplication, the dialog continues:

```
Erase flash device before writing? [confirm] n
File `abc/igs-kf.914' already exists; it will be invalidated!
Invalidate existing copy of `abc/igs-kf' in flash memory? [confirm] y
Copy `abc/igs-kf.914' from TFTP server
as `abc/igs-kf.914' into Flash WITHOUT erase? y
```

If the configuration was modified but not saved, you are asked to save the configuration:

```
System configuration has been modified. Save? [confirm]
```

If you confirm to save the configuration, you might also receive this message:

```
Warning: Attempting to overwrite an NVRAM configuration previously written by a
different version of the system image. Overwrite the previous NVRAM configuration?
[confirm]
```

Users with open Telnet connections are notified of the system reload, as:

```
**System going down for Flash upgrade**
```

If the TFTP process fails, the copy operation is retried up to thre times. If the failure happens in the middle of a copy operation so only part of the file has been written to Flash memory, the retry does not erase Flash memory unless you specified an erase operation. Th partly written file is marked as deleted, and a new file is opened with the same name. If Flash memory runs out of free space in this process, the copy operation terminates.

After Flash load helper finishes copying (whether or not the copy operation is successful), it automatically attempts an automatic or a manual boot, depending on the value o f bit0 of the configuration registe boot field:

● Bit 0 = 0	The system attempts a default boot from Flash memory to load the first bootable file in Flash memory. This default boot is equivalent to a manual **`b flash`** command at the ROM monitor prompt.
● Bit 0 = 1	The system attempts to boot based on the boot configuration commands. If no boot configuration commands exist, the system attempts a default boot from Flash memory; that is, it attempts to load the first bootable file in Flash memory.

Monitor Flash Load Helper

To view the system console output generated during the Flash load helper operation, use the image booted up after the Flash memory upgrade. Type this command in privileged EXEC mode:

```
show flh-log
```

If you are a remote Telnet user performing the Flash upgrade without a console connection, this task allows you to retrieve console output when your Telnet connection terminates due to the switch to the ROM image. The output indicates events occurring during the download and is particularly useful if the download fails.

Manually Load a System Image from ROM Monitor

If your router does not find a valid system image, or if its configuration file is corrupted at startup, and if the configuration register is set to enter ROM monitor mode, the system enters 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 router and then pressing <Break> during the first 60 seconds of startup.

Manually Boot from Flash

To manually boot from Flash memory:

1. Restart the router

   ```
   reload
   ```

2. Press <Break> during the first 60 seconds of system startup.

3. Manually boot the router from Flash.

   ```
   boot flash [filename]
   ```

In this example, a router is manually booted from Flash memory. Because the optional filename argument is absent, the first file in Flash memory is loaded.

```
> boot flash
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory RRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK - 1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

           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
```

In this example, the boot flash command is used with the filename gs7-k; the name of the file that is loaded:

```
> boot flash gs7-k
F3: 1858656+45204+166896 at 0x1000

Booting gs7-k from flash memory
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRR [OK - 1903912/13765276 bytes]
F3: 1858676+45204+166896 at 0x1000

             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
System Bootstrap, Version 4.6(1012) [mlw 99], INTERIM SOFTWARE Copyright (c) 1986-
1992 by cisco Systems RP1 processor with 16384 Kbytes of memory
```

Manually Boot from a Network File

To manually boot from a network file, complete these tasks in EXEC mode:

1. Restart the router

   `reload`

2. Press <Break> during the first 60 seconds of system startup.

3. Manually boot the router from a network file.

   boot *filename* [*ip-address*]

In this example, a router is manually booted from the network fil network1:

```
>boot network1
```

Manually Boot from ROM

To manually boot the router from ROM, complete these steps in EXEC mode:

1. Restart the router

   reload

2. Press <Break> during the first 60 seconds of system startup.
3. Manually boot the router from ROM.

   boot

In this example, a router is manually booted from ROM:

```
>boot
```

Use the System Image Instead of Reloading

To return to EXEC mode from the ROM monitor to use the system image instead of reloading, type the following in ROM monitor mode:

continue

Chapter 8: AI2524 Protocol Configuration Steps

Introduction

This chapter describes the AI2524 protocol configuration steps for th OSPF TCP/IP, IGRP TCP/IP and RIP TCP/IP.

AI2524 OSPF TCP/IP Configuration Steps

Open Shortest Path First (OSPF) is an IGP designed expressly for IP networks. OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending/receiving packets.

OSPF typically requires coordination among many internal routers, area border routers (routers connected to multiple areas), and autonomous system boundary routers. At a minimum, OSPF-based routers or access servers can be configured with all default parameter values, no authentication, and interfaces assigned to areas. If you plan to customize your environment, ensure coordinated configurations of all routers.

To configure OSPF, complete these tasks. You must enable OSPF; the other tasks are optional, but might be required for your application.

- Enable OSPF
- Configure OSPF Interface Parameters
- Configure OSPF over Different Physical Networks
- Configure OSPF Area Parameters
- Configure OSPF Not So Stubby Area (NSSA
- Configure Route Summarization between OSPF Areas
- Configure Route Summarization When Redistributing Routes into OSPF
- Create Virtual Links
- Generate a Default Route
- Configure Lookup of DNS Names
- Force the Router ID Choice with a Loopback Interfac
- Disable Default OSPF Metric Calculation Based on Bandwidth
- Configure OSPF on Simplex Ethernet Interfaces

- Configure Route Calculation Timers
- Configure OSPF over On-Demand Circuits

Enable OSPF

To enable OSPF, you must create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range of IP addresses.

1. Enable OSPF routing. This places you in the router configuration mode.

   ```
   router ospf process-id
   ```

2. Define an interface on which OSPF runs and define the area ID for that interface.

   ```
   network address wildcard-mask area area-id
   ```

Configure OSPF Interface Parameters

OSPF implementation allows you to alter certain interface-specific OSPF parameters. You may need to change these parameters for consistency. If you do configure any of these parameters, be sure the configurations for all routers on your network have compatible values.

In interface configuration mode, specify any of these interface parameters as needed for your network:

1. Explicitly specify the cost of sending a packet on an OSPF interface.

   ```
   ip ospf cost cost
   ```

2. Specify the number of seconds between link state advertisement retransmissions for adjacencies belonging to an OSPF interface.

   ```
   ip ospf retransmit-interval seconds
   ```

3. Set the estimated number of seconds it takes to transmit a link stat update packet on an OSPF interface.

   ```
   ip ospf transmit-delay seconds
   ```

4. Set priority to help determine the OSPF designated router for network.

   ```
   ip ospf priority number
   ```

5. Specify the length of time, in seconds, between the hello packets that the Cisco IOS software sends on an OSPF interface.

   ```
   ip ospf hello-interval seconds
   ```

6. Set the number of seconds that a device's hello packets must not have been seen before its neighbors declare the OSPF route down.

   ```
   ip ospf dead-interval seconds
   ```

7. Assign a specific password to be used by neighboring OSPF routers on a network segment that is using OSPF's simple password authentication.

   ```
   ip ospf authentication-key key
   ```

8. Enable OSPF MD5 authentication.

   ```
   ip ospf message-digest-key key id md5 key
   ```

Configure OSPF over Different Physical Networks

OSPF classifies different media into three types of networks by de fault:

- Broadcast networks (Ethernet, Token Ring, FDDI)
- Nonbroadcast multi-access networks (SMDS, Frame Relay, X.25
- Point-to-point networks (HDLC, PPP)

Configure your network as either a broadcast or a nonbroadcast multi-access network.

X.25 and Frame Relay provide an optional broadcast capability that can be configured in the map to allow OSPF to run as a broadcast network.

Configure OSPF for Nonbroadcast Networks

Because there might be many routers attached to an OSPF network, a designated router is selected for the network. It is necessary to use special configuration parameters in the designated router selection if broadcast capability is not configured.

These parameters need only be configured in those devices that are themselves eligible to become the designated router or backup designated router

To configure routers that interconnect to nonbroadcast networks, type this command in router configuration mode:

```
   neighbor ip-address [priority number]
[poll-interval  seconds]
```

You can specify the following neighbor parameters, as required:

- Priority for a neighboring route
- Nonbroadcast poll interval
- Interface through which the neighbor is reachable

Configure OSPF Area Parameters

OSPF software allows you to configure several area parameters. Thes area parameters, shown in the following list, include enabling authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication allows password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the area border router, into the stub area for destinations outside the autonomous system. To further reduce the number of link state advertisements sent into a stub area, you can configur **no-summary** on the Area Border Router (ABR) to prevent it from sending summary link advertisement (link state advertisements Type 3) into the stub area.

In router configuration mode, specify any of these area parameters as needed for your network:

- Enable authentication for an OSPF area.

  ```
  area area-id authentication
  ```

- Enable MD5 authentication for an OSPF area.

  ```
  area area-id authentication message-digest
  ```

- Define an area to be a stub area.

  area *area-id* stub [no-summary]

- Assign a specific cost to the default summary route used for th stub area.

  area *area-id* default-cost *cost*

Configure OSPF Not So Stubby Area (NSSA)

NSSA area is similar to OSPF stub area. NSSA does not flood Type 5 external link state advertisements (LSAs) from the core into the area, but it can import AS external routes in a limited fashion within the area.

NSSA allows importing of Type 7 AS external routes within NSSA area by redistribution. These Type 7 LSAs are translated into Type 5 LSAs by NSSA Area Border Router (ABR), which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

Use NSSA to simplify administration if you are an Internet service provider (ISP) or a network administrator that must connect a central site using OSPF to a remote site that is using a different protocol.

Prior to NSSA, the connection between the corporate site border router and the remote router could not be run as OSPF stub area because routes for the remote site could not be redistributed into stub area. A simple protocol like RIP was usually run to handle the redistribution. This meant maintaining two routing protocols. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

In router configuration mode, specify these area parameters as needed to configure OSPF NSSA:

area *area-id* nssa [no-redistribution]
[default-information-originate]

In router configuration mode on the ABR, specify this command to control summarization and filtering of Type 7 LSA into Type 5 LSA (optional):

summary address prefix mask [not advertise]
[tag tag]

Implementation Considerations

Before implementing this feature, consider these items:

- You can set a Type 7 default route that can be used to reach external destinations. When configured, the router generates a Type 7 default into the NSSA by the NSSA ABR.
- Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate with each other.
- If possible, avoid using explicit redistribution on NSSA ABR because confusion may result over which packets are being translated by which router.

Configure Route Summarization between OSPF Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an ABR. In OSPF, an ABR will advertise networks in one area into another area. If the network numbers in an area are assigned sequentially, you can configure the ABR to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To specify an address range, type this command in router configuration mode:

```
area area-id range address mask
```

Configure Route Summarization When Redistributing Routes into OSPF

When redistributing routes from other protocols into OSPF, each route is advertised individually in an external link state advertisement (LSA). However, you can configure the Cisco IOS software to adver tise a single route for all the redistributed routes covered by a specified network address and mask. Doing so helps decrease the size of th OSPF link state database.

To have the software advertise one summary route for all redistributed routes covered by a network address and mask, type this command in router configuration mode:

```
summary-address address mask
```

Create Virtual Links

In OSPF, all areas must be connected to a backbone area. If there is break in backbone continuity, or if the backbone is purposefully partitioned, you can establish a virtual link.

To establish a virtual link, type this command in router configuration mode:

```
area area-id virtual-link router-id [hello-
interval seconds] [retransmit-interval
seconds] [transmit-delay seconds]
[deadinterval seconds] [[authentication-key
key] | [message-digest-key keyid md5 key]]
```

To display information about virtual links, use th **show ip ospf virtual-links** EXEC command. To display the router ID of an OSPF router, use the **show ip ospf** EXEC command.

Generate a Default Route

You can force an autonomous system boundary router to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, th router automatically becomes an autonomous system boundary router. However, an autonomous system boundary router does not, by default, generate a default route into the OSPF routing domain.

To force the autonomous system boundary router to generate a default route, type this command in router configuration mode:

```
default-information originate [always]
[metric metricvalue] [metric-type type-
value] [route-map map-name]
```

Configure Lookup of DNS Names

You can configure OSPF to look up Domain Naming System (DNS) names for use in all OSPF show command displays. This featur makes it easier to identify a router, because it is displayed by nam rather than by its router ID or neighbor ID.

To configure DNS name lookup, type this command in global configuration mode

```
ip ospf name-lookup
```

Force the Router ID Choice with a Loopback Interface

OSPF uses the largest IP address configured on the interfaces as its router ID. If the interface associated with this IP address is ever brought down, or if the address is removed, the OSPF process must recalculate a new router ID and resend all its routing information over its interfaces.

If a loopback interface is configured with an IP address, the Cisco IOS software will use this IP address as its router ID, even if other inter faces have larger IP addresses. Since loopback interfaces never go down, greater stability in the routing table is achieved.

To configure an IP address on a loopback interface, perform thes tasks, starting in global configuration mode:

1. Create a loopback interface, which places you in interface configuration mode

   ```
   interface loopback 0
   ```

2. Assign an IP address to this interface.

   ```
   ip address address mask
   ```

Disable Default OSPF Metric Calculation Based on Bandwidth

OSPF calculates the OSPF metric for an interface according to the bandwidth of the interface. To disable this feature, type this command in router configuration mode:

```
no ospf auto-cost-determination
```

Configure OSPF on Simplex Ethernet Interfaces

Because simplex interfaces between two devices on an Ethernet represent only one network segment, for OSPF you must configure the transmitting interface to be a passive interface. This prevents OSPF from sending hello packets for the transmitting interface. Both devices are able to see each other via the hello packet generated for the receiving interface.

To configure OSPF on simplex Ethernet interfaces, type this command in router configuration mode:

```
passive-interface type number
```

Configure Route Calculation Timers

You can configure the delay time between when OSPF receives a topology change and when it starts a Shortest Path First (SPF) calculation. You can also configure the hold time between two consecutive SPF calculations. Type this command in router configuration mode:

```
timers spf spf-delay spf-holdtime
```

Configure OSPF over On-Demand Circuits

The OSPF on-demand circuit is an enhancement to the OSPF protocol that allows efficient operation over on-demand circuits like ISDN, X.25 SVCs, and dial-up lines. This feature supports RFC 1793, extending OSPF to Support Demand Circuits.

With this feature, periodic hellos are suppressed and the periodic refreshes of LSAs are not flooded over the demand circuit.

To configure OSPF for on-demand circuits, perform these tasks:

1. Enable OSPF operation.

   ```
   router ospf process-id
   ```

2. Configure OSPF on an on-demand circuit.

   ```
   ip ospf demand-circuit
   ```

If the router is part of a point-to-point topology, then only one end of the demand circuit must be configured with this command. However, all routers must have this feature loaded.

If the router is part of a point-to-multipoint topology, only the multipoint end must be configured with this command.

Implementation Considerations

Because LSAs that include topology changes are flooded over an on-demand circuit, it is advised to put demand circuits within OSPF.

Network Illustration

This list outlines key features supported in OSPF implementation:

Stub areas	Definition of stub areas is supported.
Route redistribution	Routes learned via any IP routing protocol can be redistributed into any other IP routing protocol. At the intradomain level, this means that OSPF can import routes learned via IGRP, RIP, and IS-IS. OSPF routes can also be exported into IGRP, RIP, and IS-IS. At the interdomain level, OSPF can import routes learned via EGP and BGP. OSPF routes can be exported into EGP and BGP.
Authentication	Simple and MD5 authentication among neighboring routers within an area is supported.
Routing/Interface parameters	Configurable parameters supported includ interface output cost, retransmission interval, interface transmit delay, router priority, router "dead" and hello intervals, and authentication key.
Virtual links	Virtual links are supported.
NSSA areas	RFC 1567
OSPF over demand circuit	RFC 1793

Note: ***To take advantage of the OSPF stub area support, default routing must be used in the stub area.***

AI2524 IGRP TCP/IP Configuration Steps

The Interior Gateway Routing Protocol (IGRP) is a dynamic distance-vector routing protocol for an autonomous system that contains large, arbitrarily complex networks with diverse bandwidth and delay characteristics.

IGRP uses a combination of user-configurable metrics, including in ternetwork delay, bandwidth, reliability, and load.

IGRP Updates

By default, a router running IGRP sends an update broadcast every 90 seconds. It declares a route inaccessible if it does not receive an update from the first router in the route within 3 update periods (270 seconds). After 7 update periods (630 seconds), the Cisco IOS software removes the route from the routing table.

IGRP uses flash update and poison reverse updates to speed up th convergence of the routing algorithm. Flash updates are sent soone than the standard periodic update for notifying other routers of a metric change. Poison reverse updates are intended to defeat larger routing loops caused by increases in routing metrics. The poison reverse updates are sent to remove a route and place it in holddown, which keeps new routing information from being used for a certain period of time.

IGRP Configuration Task List

To configure IGRP, perform the tasks outlined in the next section. Creating the IGRP routing process is mandatory; the other tasks are op tional.

- Create the IGRP Routing Process
- Allow Point-to-Point Updates for IGRP
- Define Unequal-Cost Load Balancing
- Control Traffic Distribution
- Adjust the IGRP Metric Weights
- Disable Holddown
- Enforce a Maximum Network Diamete
- Validate Source IP Addresses

Create the IGRP Routing Process

To create the IGRP routing process, perform these required tasks starting in global configuration mode:

1. Enable an IGRP routing process, which places you in router configuration mode.

   ```
   router igrp process number
   ```

2. Associate networks with an IGRP routing process.

   ```
   network network-number
   ```

IGRP sends updates to the interfaces in the specified networks. If an interface's network is not specified, it will not be advertised in any IGRP update.

It is not necessary to have a registered autonomous system number to use IGRP. If you do not have a registered number, you can create your own. We recommend that if you do have a registered number, you use it to identify the IGRP process.

Allow Point-to-Point Updates for IGRP

Because IGRP is normally a broadcast protocol, in order for IGRP routing updates to reach point-to-point or nonbroadcast networks, you must configure the Cisco IOS software to permit this exchange of routing information.

To permit information exchange, define a neighboring router by typing this command in router configuration mode:

```
neighbor ip-address
```

To control the set of interfaces with which to exchange routing updates, you can disable the sending of routing updates on specified in terfaces by configuring th **`passive-interface`** command.

Define Unequal-Cost Load Balancing

IGRP can simultaneously use an asymmetric set of paths for a given destination. This feature is known as unequal-cost load balancing. Unequal-cost load balancing allows traffic to be distributed among up to four unequal-cost paths to provide greater overall output and reliability. Alternate path variance (that is, the difference in desirability be tween the primary and alternate paths) is used to determine the feasibility of a potential route. Only paths that are feasible can be used for load balancing and are included in the routing table. These conditions limit the number of cases in which load balancing can occur, but ensure that the dynamics of the network will remain stable.

These general rules apply to IGRP unequal-cost load balancing:

- IGRP will accept up to four paths for a given destination network.
- The local best metric must be greater than the metric learned from the next router; that is, the next-hop router must be closer (have a smaller metric value) to the destination than the local best metric.
- The alternative path metric must be within the specified variance of the local best metric. The multiplier times the local best metric for the destination must be greater than or equal to the metric through the next router

If these conditions are met, the route is deemed feasible and can b added to the routing table.

By default, the amount of variance is set to one (equal-cost load balancing). You can define how much worse an alternate path can be before that path is disallowed by defining the variance associated with a particular path variance multiplier.

Note: ***By using the variance feature, the Cisco IOS software can balance traffic across all feasible paths and can immediately converge to a new path if one of the paths fails.***

Control Traffic Distribution

By default, if IGRP or Enhanced IGRP have multiple routes of unequal cost to the same destination, the Cisco IOS software will distribute traffic among the different routes by giving each route a share o the traffic in inverse proportion to its metric. If you want to have faster convergence to alternate routes, but you do not want to send traffic across inferior routes in the normal case, you might prefer to have no traffic flow along routes with higher metrics.

To control how traffic is distributed among multiple routes of unequal cost, type this command in router configuration mode:

```
traffic-share {balanced | min}
```

Adjust the IGRP Metric Weights

You can alter the default behavior of IGRP routing and metric computations. Although IGRP metric defaults were carefully selected to provide excellent operation in most networks, you can adjust the IGRP metric. Adjusting IGRP metric weights can dramatically affect network performance, however, so ensure that you make all metric adjustments carefully.

Note: ***Because of the complexity of this task, we recommend that you only perform it with guidance from an experienced system designer.***

To adjust the IGRP metric weights, type this command in router configuration mode:

```
metric weights tos k1 k2 k3 k4 k5
```

By default, the IGRP composite metric is a 24-bit quantity that is a sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Ethernet, and serial lines running from 9600 bps to T1 rates), the route with the lowest metric reflects the most desirable path to destination.

Disable Holddown

When the Cisco IOS software learns that a network is at a greater distance than was previously known, or it learns the network is down, th route to that network is placed in holddown. During the holddown period, the route is advertised, but incoming advertisements about that network from any router other than the one that originally advertised the network's new metric will be ignored. All devices in an IGRP autonomous system must be consistent in their use of holddowns. To disable holddowns with IGRP, type this command in router configuration mode:

```
no metric holddown
```

Enforce a Maximum Network Diameter

The Cisco IOS software enforces a maximum diameter to the IGRP network. Routes whose hop counts exceed this diameter are not advertised. The default maximum diameter is 100 hops. The maximum diameter is 255 hops.

To configure the maximum diameter, type this command in router configuration mode:

```
metric maximum-hops hops
```

Validate Source IP Addresses

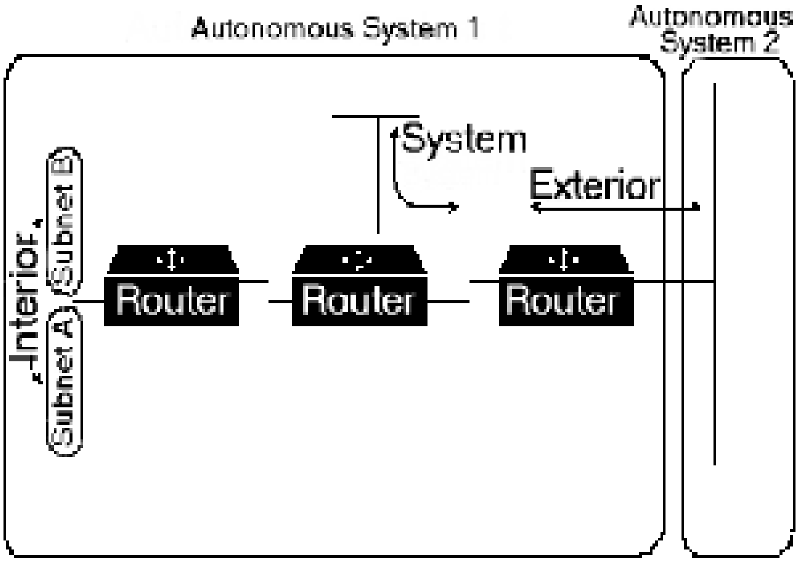
To disable the default function that validates the source IP addresses of incoming routing updates, type this command in router configura tion mode:

```
no validate-update-source
```

Network Illustration

IGRP advertises three types of routes: interior, system, and exterior, as shown in Figure 8-1. Interior routes are routes between subnets in the network attached to a router interface. If the network attached to a router is not subnetted, IGRP does not advertise interior routes.

Figure 8-1:Interior, System, and Exterior Routes



System routes are routes to networks within an autonomous system. The Cisco IOS software derives system routes from directly connected network interfaces and system route information provided by other IGRP-speaking routers or access servers. System routes do not include subnet information.

Exterior routes are routes to networks outside the autonomous system that are considered when identifying a gateway of last resort. Th Cisco IOS software chooses a gateway of last resort from the list of exterior routes that IGRP provides. The software uses the gateway (router) of last resort if it does not have a better route for a packet and the destination is not a connected network. If the autonomous system has more than one connection to an external network, different routers can choose different exterior routers as the gateway of last resort.

AI2524 RIP TCP/ IP Configuration

Routing Information Protocol (RIP) uses broadcast User Datagram Protocol (UDP) data packets to exchange routing information.

The Cisco IOS software sends routing information updates (advertisements) every 30 seconds. If a router does not receive an update from another router for 180 seconds or more, it marks the routes served by the nonupdating router as being unusable. If there is still no update after 240 seconds, the router removes all routing table entries for the nonupdating router.

RIP sends updates to the interfaces in the specified networks. If an interface's network is not specified, it will not be advertised in any RIP update.

RIP Configuration Task List

To configure RIP, complete the tasks outlined in the nest sections. You must enable RIP. The remaining tasks are optional.

- Enable RIP
- Allow Point-to-Point Updates for RI
- Specify a RIP Version
- Enable RIP Authentication
- Disable Route Summarization
- Run IGRP and RIP Concurrently
- Disable the Validation of Source IP Addresses

Enable RIP

To enable RIP, perform these tasks, starting in global configuration mode:

1. Enable a RIP routing process, which places you in router configuration mode.

   ```
   router rip
   ```

2. Associate a network with a RIP routing process.

   ```
   network network-number
   ```

Allow Point-to-Point Updates for RIP

RIP is normally a broadcast protocol. Therefore, for RIP routing updates to reach point-to-point or nonbroadcast networks, you must configure the Cisco IOS software to permit this exchange of routing information. To define a neighboring router with which to exchange point-to-point routing information, type this command in router configuration mode:

```
neighbor ip-address
```

To control the set of interfaces with which to exchange routing updates, you can disable the sending of routing updates on specified in terfaces by configuring the passive interface command.

Specify a RIP Version

By default, the software receives RIP Versi on1 and Vers ion2 packets, but sends only Vers ion1 packets. You can configure the softwar to receive and send only Ver sion1 packets. Alternatively, you can configure the software to receive and send only Vers ion2 packets. To configure the software to receive and send only RIP Ver sion1 or only RIP Version 2 packets, type this command in router configuration mode:

```
version {1 | 2}
```

The preceding task controls the default behavior of RIP. You can override that behavior by configuring a particular interface to behave differently. To control which RIP version an interface sends, perform one of these tasks in interface configuration mode:

- Configure an interface to send only RIP Ver sion1 packets.

  ```
  ip rip send version 1
  ```

- Configure an interface to send only RIP Ver sion2 packets.

  ```
  ip rip send version 2
  ```

- Configure an interface to send RIP Ver sion1 and Ve rsion2 packets.

  ```
  ip rip send version 1 2
  ```

Similarly, to control how packets received from an interface are processed, perform one of these tasks in interface configuration mode:

- Configure an interface to accept only RIP Ver sion1 packets.

  ```
  ip rip receive version 1
  ```

- Configure an interface to accept only RIP Ver sion2 packets.

  ```
  ip rip receive version 2
  ```

- Configure an interface to accept either RIP Ve rsion1 or 2 packets.

  ```
  ip rip receive version 1 2
  ```

Enable RIP Authentication

RIP Version 1 does not support authentication. If you are sending and receiving RIP Versio n2 packets, you can enable RIP authentication on an interface.

The key chain determines the set of keys that can be used on the interface. If a key chain is not configured, no authentication is performed on that interface, not even the default authentication. Therefore, you must also perform the tasks in the section Manage Authentication Keys later in this chapter.

The software supports two modes of authentication on an interface for which RIP authentication is enabled: plain text authentication and MD5 authentication. The default authentication in every RIP Version 2 packet is plain text authentication.

Note: ***For security purposes, do not use plain text authentication in RIP packets, because the unencrypted authentication key is sent in every RIP Versio n2 packet. Use plain text authentication when security is not an issue, for example, to ensure that misconfigured hosts do not participate in routing.***

To configure RIP authentication, perform these tasks in interface configuration mode:

1. Enable RIP authentication.

```
ip rip authentication key-chain name-of-chain
```

2. Configure the interface to use MD5 digest authentication (or let it default to plain text authentication).

```
ip rip authentication mode {text | md5}
```

Disable Route Summarization

RIP Version 2 supports automatic route summarization by default. The software summarizes subprefixes to the classful network boundary when crossing classful network boundaries.

If you have disconnected subnets, disable automatic route summarization to advertise the subnets. When route summarization is disabled, the software transmits subnet and host routing information across classful network boundaries. To disable automatic summarization, type this command in router configuration mode:

```
no auto-summary
```

Run IGRP and RIP Concurrently

It is possible to run IGRP and RIP concurrently. The IGRP information will override the RIP information by default because of IGRP's administrative distance.

Running IGRP and RIP concurrently does not work well when the network topology changes. Because IGRP and RIP have different update timers, and because they require different amounts of time to propagate routing updates, one part of the network will follow IGRP routes and another part will follow RIP routes. This results in routing loops. Although these loops do not exist for long, the Time To Live (TTL) will quickly reach zero, and ICMP will send a TTL exceeded message. This message will cause most applications to stop attempting network connections.

Disable the Validation of Source IP Addresses

By default, the software validates the source IP address of incoming RIP routing updates. If that source address is not valid, the software discards the routing update.

Consider disabling this feature if you have a router that is off network and you want to receive its updates. Disabling this feature is not rec ommended under normal circumstances. To disable the default func tion that validates the source IP addresses of incoming routing updates, perform this task in router configuration mode:

- Disable the validation of the source IP address of incoming RIP routing updates.

  ```
  no validate-update-source ion
  ```

In addition to running multiple routing protocols simultaneously, the Cisco IOS software can redistribute information from one routing protocol to another. For example, you can instruct the software to readvertise IP Enhanced IGRP-derived routes using the RIP protocol or to readvertise static routes using the IP Enhanced IGRP protocol. This capability applies to all the IP-based routing protocols.

You may also conditionally control the redistribution of routes be tween routing domains by defining a method known as route maps between the two domains.

- To redistribute routes from one protocol into another, type this command in router configuration mode:

  ```
  redistribute protocol autonomous-system-
  number [route-map map-tag]
  ```

- To define route maps, type this command in global configuration mode:

  route-map ***map-tag*** **[permit | deny]**
  [*sequence-number*]

- By default, the redistribution of default information between IP Enhanced IGRP processes is enabled. To disable the redistribution, type this command in router configuration mode:

  no default-information {in | out}

Chapter 9: AI2524 OSI/CLNP Configuration Steps

Introduction

This chapter describes how to configure ISO CLNS. The ISO CLNS protocol is a standard for the network layer of the OSI model.

ISO CLNS Configuration Task List

To configure ISO CLNS, you must configure the routing processes, associate addresses with the routing processes, and customize the routing processes for your particular network.

You must perform some combination of the tasks listed here to configure the ISO CLNS protocol:

- Understand Addresses
- Understand Routing Processes
- Configure ISO IGRP Dynamic Routing
- Configure IS-IS Dynamic Routing
- Configure CLNS Static Routing
- Configure Miscellaneous Features
- Configure CLNS over WANs
- Enhance ISO CLNS Performance
- Monitor and Maintain the ISO CLNS Network
- Configure TARP on ISO CLNS

Understand Addresses

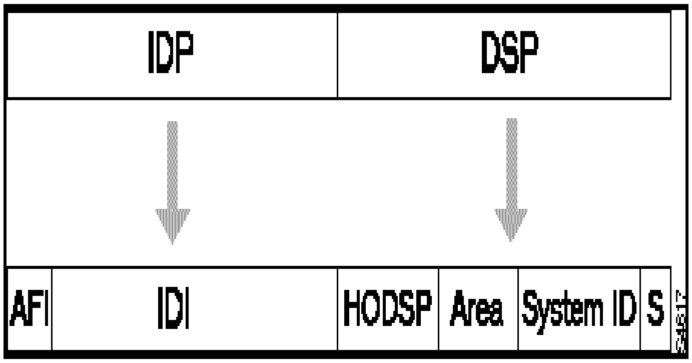
Addresses in the ISO network architecture are referred to as NSAP addresses and network entity titles (NETs). Each node in an OSI network has one or more NETs. In addition, each node has many NSAP addresses. Each NSAP address differs from one of the NETs for that node in only the last byte. This byte is called the N-selector. Its function is similar to the port number in other protocol suites.

The AI2524 router supports all NSAP address formats that are defined by ISO 8348/Ad2; however, the AI2524 router provides ISO IGRP o IS-IS dynamic routing only for NSAP addresses that conform to the address constraints defined in the ISO standard for IS-IS (ISO 10589).

An NSAP address consists of these two major fields, as shown inFigure 9-1:

- The initial domain p art (IDP) is made up of 1-byte authority and format identifier (AFI) and a variable-length initial domai nidentifier (IDI). The length of the IDI and the encoding format for the domai nspecific part (DSP) are based on the value of the AFI.
- The DSP is made up of a high-order DSP, an area identifier, a system identifier, and a 1-byte N-selector (labeled S).

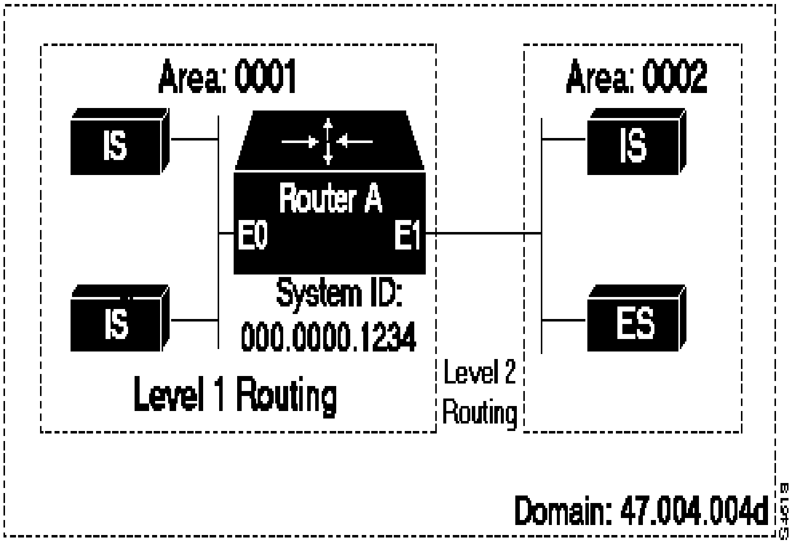
Figure 9-1:NSAP Address Fields



Assign addresses or NETs for your domains and areas. Th domai naddress uniquely identifies the routing domain. All routers within a given doma inare given the same dom ainaddress. Within each routing domain, you can set up one or more areas, as shown in Figure 9-2. Determine which routers are to be assigned to which areas.

The area address uniquely identifies the routing area and the system ID identifies each node.

Figure 9-2:Sample Domain and Area Addresses



The key difference between the ISO IGRP and IS-IS NSAP addressing schemes is in the definition of area addresses. Both use the system ID for Level 1 routing (routing within an area). However, they differ in the way addresses are specified for area routing. An ISO IGRP NSAP address includes three separate fields for routing: the domain, area, and system ID. An IS-IS address includes two fields: a single contin uous area field (comprising the doma inand area fields) and the system ID.

ISO IGRP NSAP Address

The ISO IGRP NSAP address is divided into three parts: a domai npart, an area address, and a system ID. D omainrouting is performed on the doma inpart of the address. Area routing for a given domai nuses the area address. System routing for a given area uses the system ID part. The NSAP address is laid out as:

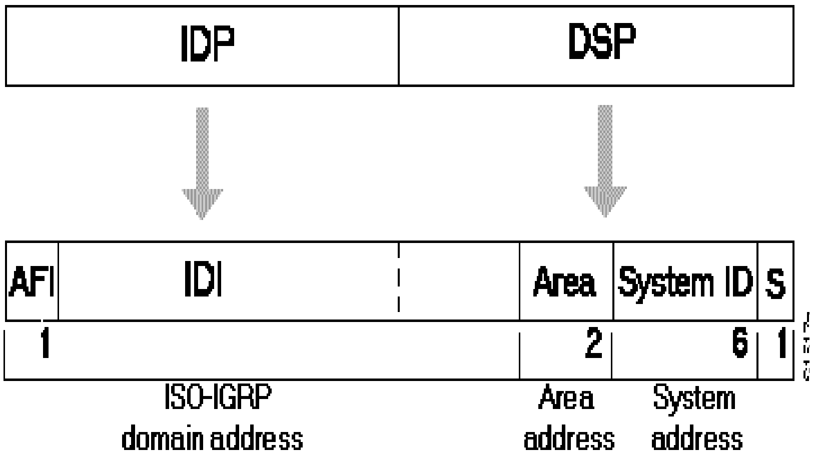
- The domai npart is of variable length and comes before the area address.
- The area address is the 2 bytes before the system ID.

- The system ID is the 6 bytes before the N-selector.
- The N-selector (S) is the last byte of the NSAP address.

Cisco's ISO IGRP routing implementation interprets the bytes from the AFI up to (but not including) the area field in the DSP as domai nidentifier. The area field specifies the area, and the system ID specifies the system.

Figure 9-3 illustrates the ISO IGRP NSAP addressing structure. The maximum address size is 20 bytes.

Figure 9-3:ISO IGRP NSAP Addressing Structure



IS-IS NSAP Address

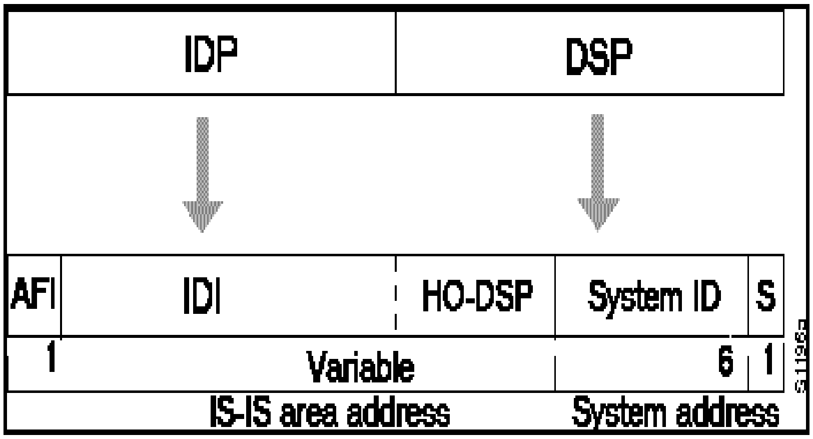
An IS-IS NSAP address is divided into two parts: an area address and a system ID. Level 2 routing (routing between areas) uses the area address. Leve l1 routing (routing within an area) uses the system ID address. The NSAP address is laid out as:

- The area address is the NSAP address, not including the system ID and N-selector.
- The system ID is found between the area address and the N-selector byte.
- The N-selector (S) is the last byte of the NSAP address.

The IS-IS routing protocol interprets the bytes from the AFI up to (but not including) the system ID field in the DSP as an area identifier. Th system ID specifies the system.

Figure 9-4 illustrates the IS-IS NSAP addressing structure. The maximum address size is 20 bytes.

Figure 9-4:IS-IS NSAP Addressing Structure



Addressing Rules

All NSAP addresses must obey these constraints:

- No two nodes can have addresses with the same NET; that is, addresses that match all but the N-selector (S) field in the DSP.
- No two nodes residing within the same area can have addresses in which the system ID fields are the same.
- ISO IGRP requires at least 10 bytes of length: 1 byte for domain, 2 bytes for area, 6 bytes for system ID, and 1 byte for N-selector.
- ISO IGRP and IS-IS should not be configured for the same area. Do not specify an NSAP address where all bytes up to (but not including) the system ID are the same when enabling both ISO IGRP and IS-IS routing.
- A router can have one or more area addresses. The concept of multiple area addresses is described in Assign Multiple Area Addresses to IS-IS Areas.
- IS-IS requires at least 8 bytes: one byte for area, 6 bytes for system ID, and 1 byte for N-selector.

Addressing Examples

Examples of OSI network and GOSIP NSAP addresses using the ISO IGRP implementation are described in the next sections.

The OSI network NSAP address format is illustrated as:

```
|     Domain|Area|     System ID| S| 47.0004.004D.0003.0000.0C00.62E6.00
```

This is an example of the GOSIP NSAP address structure. This structure is mandatory for addresses allocated from the International Code Designator (ICD) 0005 addressing domain. Refer to the GOSIP document, U.S. Government Open Systems Interconnection Profile (GOSIP), Draft Version 2.0, April 1989, for more information.

```
|                    Domain|     Area|System ID| S|
47.0005.80.ffff00.0000.ffff.0004.0000.0c00.62e6.00
 |   |    |    |    |
AFI IDI  DFI  AAI Resv  RD
```

Routing Table Example

You enter static routes by specifying NSAP prefix and next-hop NET pairs (by using the **`clns route`** command). The NSAP prefix can be any portion of the NSAP address. NETs are similar in function to NSAP addresses.

If an incoming packet has a destination NSAP address that does not match any existing NSAP addresses in the routing table, the Cisco IOS software will try to match the NSAP address with an NSAP prefix to route the packet. In the routing table, the best match means the longest NSAP prefix entry that matches the beginning of the destination NSAP address.

This table shows a sample static routing table in which the nexthop NETs are listed for completeness, but are not necessary to understand the routing algorithm.

Entry	NSAP Address Prefix	Next-Hop NET
1	47.0005.000c.0001	47.0005.000c.0001.0000.1234.00
2	47.0004	47.0005.000c.0002.0000.0231.00
3	47.0005.0003	47.0005.000c.0001.0000.1234.00
4	47.0005.000c	47.0005.000c.0004.0000.0011.00
5	47.0005	47.0005.000c.0002.0000.0231.00

This table offers examples of how the longest matching NSAP prefix can be matched with routing table entries in the table above.

Datagram Destination NSAP Address	Table Entry Number Used
47.0005.000c.0001.0000.3456.01	1
47.0005.000c.0001.6789.2345.01	1
47.0004.1234.1234.1234.1234.01	2
47.0005.0003.4321.4321.4321.01	3
47.0005.000c.0004.5678.5678.01	4
47.0005.0001.0005.3456.3456.01	5

Octet boundaries must be used for the internal boundaries of NSAP addresses and NETs.

Understand Routing Processes

The basic function of a router is to forward packets: receive a packet in one interface and send it out another (or the same) interface to the proper destination. All routers do this by looking up the destination address in a table. The tables can be built either dynamically or statically. If you are configuring all the entries in the table yourself, you are using static routing. If you use a routing process to build the tables, you ar using dynamic routing. It is possible, and sometimes necessary, to use both static and dynamic routing simultaneously.

When you configure only ISO CLNS and not routing protocols, the Cisco IOS software only makes forwarding decisions. It does not perform other routing-related functions. In such a configuration, the software compiles a table of adjacency data, but does not advertise this information. The only information that is inserted into the routing table is the NSAP and NET addresses of this router, static routes, and adjacency information.

You can route ISO CLNS on some interfaces and transparently bridge it on other interfaces simultaneously. To do this, you must enable concurrent routing and bridging by using th **`bridge crb`** command.

Dynamic Routing

Cisco supports these two dynamic routing protocols for ISO CLNP networks:

- ISO IGRP
- IS-IS

When dynamically routing, you can choose either ISO IGRP or IS-IS, or you can enable both routing protocols at the same time. Both routing protocols support the concept of areas. Within an area, all routers know how to reach all the system IDs. Between areas, routers know how to reach the proper area.

ISO IGRP supports three levels of routing: system routing, area routing, and interdomainrout ing. Routing across domains (interdomai nrouting) can be static or dynamic with ISO IGRP. IS-IS supports two levels of routing: station routing (within an area) and are routing (between areas).

Intermediate Systems (IS) and End Systems (ES)

Some ISs keep track of how to communicate with all the ESs in their areas and thereby function as Level 1 routers (also referred to as local routers). Other ISs keep track of how to communicate with other areas in the domain, functioning as Le vel2 routers (sometimes referred to as area routers). The AI2524 router is always Le vel1 and L evel2 when routing ISO IGRP; it can be configured to be Level 1 only, Level2 only, or both Level 1 and Leve l2 when routing IS-IS.

ESs communicate with ISs using the ES-IS protocol. Lev el1 and Level 2 ISs communicate with each other using either ISO IS-IS or Cisco's ISO IGRP protocol.

Static Routing

Static routing is used when it is not possible or desirable to use dynamic routing. Here are some instances of when you would use static routing:

- If your network includes WAN links that involve paying for connect time or for per-packet charges, use static routing, rather than paying to run a routing protocol and all its routing update packets over that link.
- If you want routers to advertise connectivity to external networks, but you are not running an interd o mainrouting protocol, you must use static routes.
- If you must interoperate with another vendor's equipment that does not support any of the dynamic routing protocols that Cisco supports, you must use static routing.
- For operation over X.25, Frame Relay, or SMDS networks, static routing is generally preferable.

Warning: An interface that is configured for static routing cannot reroute around failed links.

Routing Decisions

A CLNP packet sent to any of the defined NSAP addresses or NETs will be received by the router. The Cisco IOS software uses this algorithm to select which NET to use when it sends a packet:

- If no dynamic routing protocol is running, use the NET defined for the outgoing interface, if it exists; otherwise, use the NET defined for the router.
- If ISO IGRP is running, use the NET of the ISO IGRP routing process that is running on the interface.
- If IS-IS is running, use the NET of the IS-IS routing process that is running on the interface.

Configure ISO IGRP Dynamic Routing

The ISO IGRP is a dynamic distance-vector routing protocol designed by Cisco for routing an autonomous system that contains large, arbitrarily complex networks with diverse bandwidth and delay characteristics.

To configure ISO IGRP, complete the tasks outlined in these sections. Only enabling ISO IGRP is required; the remaining task is optional, although you might be required to perform it, depending upon your specific application:

- Enable ISO IGRP
- Configure ISO IGRP Parameters

In addition, you can also configure these miscellaneous features:

- Filter routing information (refer to Create Packet-Forwarding Filters and Establish Adjacencies)
- Redistribute routing information from one routing process to another (refer to Redistribute Routing Information)
- Configure administrative distances (refer to Specify Preferred Routes)

Enable ISO IGRP

To configure ISO IGRP dynamic routing, you must enable the ISO IGRP routing process, identify the address for the router, and specify the interfaces that are to route ISO IGRP. Optionally, you can set a level for your routing updates when you configure the interfaces. CLNS routing is enabled by default on routers when you configure ISO IGRP. You can specify up to ten ISO IGRP routing processes.

In global configuration mode, configure ISO IGRP dynamic routing on the router:

1. Enable the ISO IGRP routing process and enter router configuration mode.

   ```
   router iso-igrp [tag]
   ```

2. Configure the NET or address for the routing process.

   ```
   net network-entity-title
   ```

Although IS-IS allows you to configure multiple NETs, ISO IGRP allows only one NET per routing process.

You can assign a meaningful name for the routing process by using the tag option. You can also specify a name for a NET in addition to an address. For information on how to assign a name, see Specify Shortcut NSAP Addresses.

You can configure an interface to advertise L evel2 information only. This option reduces the amount of router-to-router traffic by telling th Cisco IOS software to send out only L evel2 routing updates on certain interfaces. Level 1 information is not passed on the interfaces for which the Level 2 option is set.

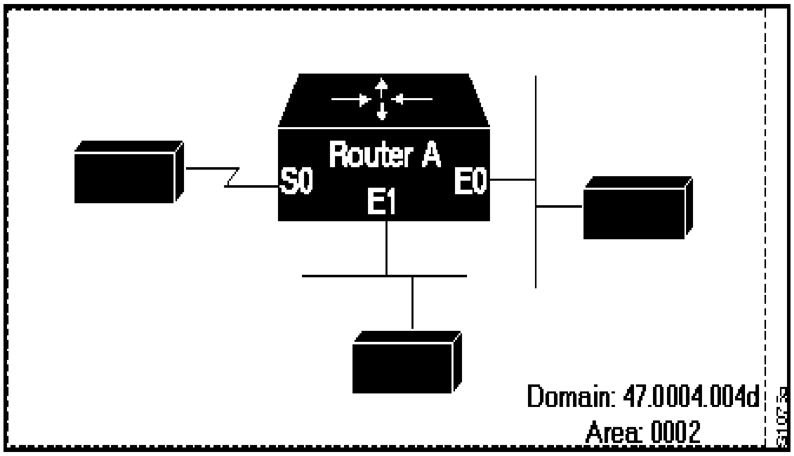
In interface configuration mode, enable ISO IGRP on specified inter faces and set the level type for routing updates:

```
clns router iso-igrp tag [level 2]
```

Example: Dynamic Routing within the Same Area

Figure 9-5, and the example configuration illustrate how to configure dynamic routing within a routing domain. The router can exist in on or more areas within the domain. The router named Router A exists in a single area:

Figure 9-5:CLNS Dynamic Routing within a Single Area



1. Define a tag castor for the routing process:

```
router iso-igrp castor
```

2. Configure the net for the process in area 2, doma in47.0004.004d:

```
net 47.0004.004d.0002.0000.0C00.0506.00
```

3. Specify iso-igrp routing using the previously specified tag castor :

```
interface ethernet 0
clns router iso-igrp castor
```

4. Specify iso-igrp routing using the previously specified tag castor :

```
interface ethernet 1
clns router iso-igrp castor
```

5. Specify iso-igrp routing using the previously specified tag castor:

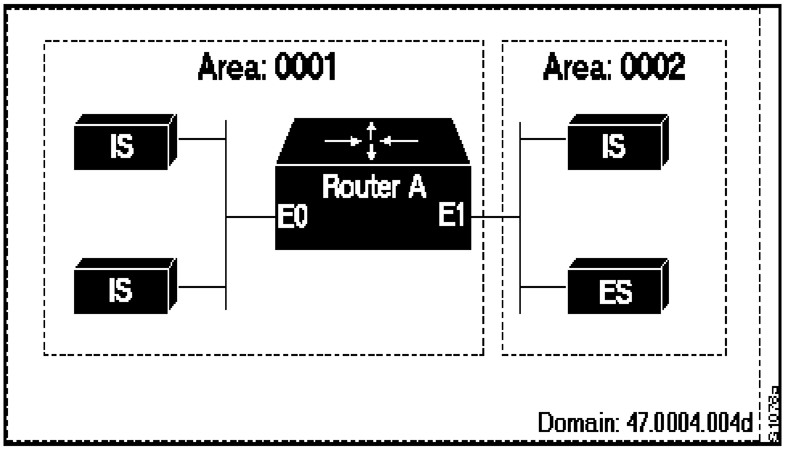
```
interface serial 0
clns router iso-igrp castor
```

Example: Dynamic Routing in More Than One Area

Figure 9-6 and the example configuration illustrate how to configure router named Router A that exists in two areas:

Figure 9-6:CLNS Dynamic Routing within Two Areas

.



1. Define a tag orion for the routing process:

```
router iso-igrp orion
```

2. Configure the net for the process in area 1, doma in47.0004.004d:

```
net 47.0004.004d.0001.212223242526.00
```

3. Specify iso-igrp routing using the previously specified tag orion:

```
interface ethernet 0
clns router iso-igrp orion
```

4. Specify iso-igrp routing using the previously specified tag orion:

```
interface ethernet 1
clns router iso-igrp orion
```

Example: Dynamic Routing in Overlapping Areas

This example illustrates how to configure a router with overlapping areas:

1. Define a tag capricorn for the routing process:

```
router iso-igrp capricorn
```

2. Configure the NET for the process in area 3,
   domai n47.0004.004d:

```
net 47.0004.004d.0003.0000.0C00.0508.00
```

3. Define a tag cancer for the routing process:

```
router iso-igrp cancer
```

4. Configure the NET for the process in area 4,
domai n47.0004.004d:

```
net 47.0004.004d.0004.0000.0C00.0506.00
```

5. Specify iso-igrp routing on interface ethernet 0 using the tag capricorn:

```
interface ethernet 0
clns router iso-igrp capricorn
```

6. Specify iso-igrp routing on interface ethernet 1 using the tags capricorn and cancer:

```
interface ethernet 1
clns router iso-igrp capricorn
clns router iso-igrp cancer
```

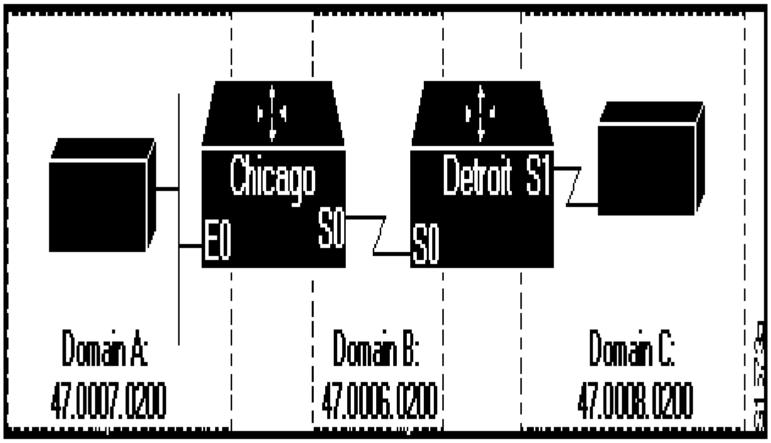
7. Specify iso-igrp routing on interface ethernet 2 using the tag cancer:

```
interface ethernet 2
clns router iso-igrp cancer
```

Example: Dynamic Interdomain R outing

Figure 9-7 and the example configurations illustrate how to configur three domains that are to be transparently connected.

Figure 9-7:CLNS Dynamic Interdomain R outing



Router Chicago

This configuration shows how to configure Router Chicago for dynamic interdomai nrouting:

1. Define a tag A for the routing process:

```
router iso-igrp A
```

2. Configure the NET for the process in area 2, domai n47.0007.0200:

```
net 47.0007.0200.0002.0102.0104.0506.00
```

3. Redistribute iso-igrp routing information througho u t domainA:

```
redistribute iso-igrp B
```

4. Define a tag B for the routing process:

```
router iso-igrp B
```

5. Configure the NET for the process in area 3, domai n47.0006.0200:

```
net 47.0006.0200.0003.0102.0104.0506.00
```

6. Redistribute iso-igrp routing information througho u t domainB:

```
redistribute iso-igrp A
```

7. Specify iso-igrp routing with the tag A:

```
interface ethernet 0
clns router iso-igrp A
```

8. Specify iso-igrp routing with the tag B:

```
interface serial 0
clns router iso-igrp B
```

Router Detroit

This configuration shows how to configure Router Detroit for dynamic interdomainrouting. Comment lines have been eliminated from this example to avoid redundancy.

```
router iso-igrp B
net 47.0006.0200.0004.0102.0104.0506.00
redistribute iso-igrp C
router iso-igrp C
net 47.0008.0200.0005.0102.01040.506.00
redistribute iso-igrp B
interface serial 0
clns router iso-igrp B
interface serial 1
clns router iso-igrp C
```

Chicago injects a prefix route for domain A into domai nB. Domai nB injects this prefix route and a prefix route for doma inB into do mainC.

You also can configure a border router between d o mainA and domai nC.

Configure ISO IGRP Parameters

Cisco's ISO IGRP implementation allows you to customize certain ISO IGRP parameters:

- Adjust ISO IGRP Metrics
- Adjust ISO IGRP Timers
- Enable or Disable Split Horizon

Adjust ISO IGRP Metrics

You have the option of altering the default behavior of ISO IGRP routing and metric computations. This allows, for example, the tuning of system behavior to allow for transmissions via satellite. Although ISO IGRP metric defaults were carefully selected to provide excellent operation in most networks, you can adjust the metric.

Warning:** **Adjusting the ISO IGRP metric can dramatically affect network performance, so ensure that all metric adjustments are made carefully. Because of the complexity of this task, it is not recommended unless it is done with guidance from an experienced system designer.

You can use different metrics for the ISO IGRP routing protocol on CLNS. In router configuration mode, configure the metric constants used in the ISO IGRP composite metric calculation of reliability and load:

```
   metric weights qos k1 k2 k3 k4 k5
```

Two additional ISO IGRP metrics can be configured: the bandwidth and delay associated with an interface.

Warning: Using the **`bandwidth`** ***and*** **`delay`** ***commands to change the values of the ISO IGRP metrics also changes the values of IP IGRP metrics.***

Adjust ISO IGRP Timers

The basic timing parameters for ISO IGRP are adjustable. Because the ISO IGRP routing protocol executes a distributed, asynchronous routing algorithm, it is important that these timers be the same for all routers in the network.

In router configuration mode, adjust ISO IGRP timing parameters:

```
   timers basic update-interval holddown-
interval invalidinterval
```

Enable or Disable Split Horizon

Split horizon blocks information about routes from being advertised out the interface from which that information originated. This feature usually optimizes communication among multiple routers, particularly when links are broken.

In interface configuration mode, either enable or disable split horizon for ISO IGRP updates:

- Enable split horizon for ISO IGRP updates.

```
   clns split-horizon
```

- Disable split horizon for ISO IGRP updates.

```
no clns split-horizon
```

The default for all LAN interfaces is for split horizon to be enabled; the default for WAN interfaces on X.25, Frame Relay, or SMDS networks is for split horizon to be disabled.

Configure IS-IS Dynamic Routing

IS-IS is a dynamic routing specification described in ISO 10589. Cisco's implementation of IS-IS allows you to configure IS-IS as an ISO CLNS routing protocol.

To configure IS-IS, complete these tasks. Only enabling IS-IS is re quired; the remainder of the tasks are optional, although you might b required to perform them depending upon your specific application.

- Enable IS-IS
- Assign Multiple Area Addresses to IS-IS Areas
- Configure IS-IS Parameters
- Configure IS-IS Interface Parameters

In addition, you can also configure these miscellaneous features:

- Filter routing information (refer to Create Packet-Forwarding Filters and Establish Adjacencies)
- Redistribute routing information from one routing process to another (refer to Redistribute Routing Information)
- Configure administrative distances (refer to Specify Preferred Routes)

Enable IS-IS

To configure IS-IS dynamic routing, you must enable the IS-IS routing process, identify the address for the router, and specify the interfaces that are to route IS-IS. CLNS routing is enabled by default when you configure IS-IS dynamic routing. You can specify only one IS-IS process per router.

In global configuration mode, configure IS-IS dynamic routing on the router:

1. Enable IS-IS routing and enter router configuration mode.

```
router isis [ tag]
```

2. Configure the NET for the routing process.

   net *network-entity-title*

You can assign a meaningful name for the routing process by using the tag option. You can also specify a name for a NET in addition to an address. For information on how to assign a name, see Specify Shortcut NSAP Addresses.

In interface configuration mode, specify the interfaces that should be actively routing IS-IS:

clns router isis [*tag*]

Warning: For IS-IS, multiple NETs per router are allowed, with a maximum of three. However, only one ISIS process is allowed, whether you run it in integrated mode, ISO CLNS only, or IP only.

Examples: IS-IS Routing Configuration

These examples illustrate the basic syntax and configuration command sequence for IS-IS routing.

Level1 and Level2 Routing

This example illustrates using the IS-IS protocol to configure a single area address for Lev el1 and Le vel2 routing:

1. Route dynamically using the IS-IS protocol:

```
router isis
```

2. Configure the NET for the process in area 47.0004.004d.0001:

```
net 47.0004.004d.0001.0000.0c00.1111.00
```

3. Enable IS-IS routing on ethernet 0:

```
interface ethernet 0 clns router isis
```

4. Enable IS-IS routing on ethernet 1:

```
interface ethernet 1
clns router isis
```

5. Enable IS-IS routing on serial 0:

```
interface serial 0
clns router isis
```

Level2 Routing Only

This example illustrates a similar configuration, featuring a single area address being used for specification of Level 1 and Level2 routing. However, in this case, interface serial interface 0 is configured for Level 2 routing only. Most comment lines have been eliminated from this example to avoid redundancy.

```
router isis
net 47.0004.004d.0001.0000.0c00.1111.00
interface ethernet 0
clns router isis
interface ethernet 1
clns router isis
interface serial 0
clns router isis
```

1. Configure a level 2 adjacency only for interface serial 0:

```
isis circuit-type level-2-only
```

OSI Configuration

This example illustrates an OSI configuration example. In this example, IS-IS runs with two area addresses, metrics tailored, and different circuit types specified for each interface. Most comment lines have been eliminated from this example to avoid redundancy.

1. Enable IS-IS routing in area 1:

```
router isis area1
```

2. Router is in areas 47.0004.004d.0001 and 47.0004.004d.0011:

```
net 47.0004.004d.0001.0000.0c11.1111.00
net 47.0004.004d.0011.0000.0c11.1111.00
```

3. Enable the router to operate as a station router and an interare router:

```
is-type level-1-2

!
interface ethernet 0
clns router isis area1
```

4. Specify a cost of 5 for the level-1 routes:

```
isis metric 5 level-1
```

5. Establish a level-1 adjacency:

```
isis circuit-type level-1
!
interface ethernet 1
clns router isis area1
isis metric 2 level-2
isis circuit-type level-2-only
!
interface serial 0
clns router isis area1
isis circuit-type level-1-2
```

6. Set the priority for serial 0 to 3 for a level-1 adjacency:

```
isis priority 3 level-
isis priority 1 level-
```

ISO CLNS Dynamic Route Redistribution

This example illustrates route redistribution between IS-IS and ISO IGRP domains. In this case, the IS-IS domain is on Ethernet interface 0; the ISO IGRP doma inis on serial interface 0. The IS-IS routing process is assigned a null tag; the ISO IGRP routing process is assigned a tag of remote-domain. Most comment lines have been eliminated from this example to avoid redundancy.

```
router isis
net 39.0001.0001.0000.0c00.1111.00
```

1. Redistribute iso-igrp routing information throughout remote-domain:

```
redistribute iso-igrp remote-domai n
!
router iso-igrp remote-domain
net 39.0002.0001.0000.0c00.1111.00
```

2. Redistribute IS-IS routing information:

```
redistribute isis
!
interface ethernet 0
clns router isis
!
interface serial 0
clns router iso-igrp remote
```

Assign Multiple Area Addresses to IS-IS Areas

IS-IS routing supports the assignment of multiple area addresses on the same router. This concept is referred to as multihoming. Multihoming provides a mechanism for smoothly migrating network addresses as:

- Splitting up an area. Nodes within a given area can accumulate to a point that they are difficult to manage, cause excessive traffic, or threaten to exceed the usable address space for an area. Multiple area addresses can be assigned so that you can smoothly partition a network into separate areas without disrupting service.
- Merging areas. Use transitional area addresses to merge as many as three separate areas into a single area that share a common area address.
- Transition to a different address. You may need to change an are address for a particular group of nodes. Use multiple area addresses to allow incoming traffic intended for an old area address to continue being routed to associated nodes.

You must statically assign the multiple area addresses on the router. You can assign up to three area addresses on the AI2524. The number of areas allowed in a doma inis unlimited.

All the addresses must have the same system ID. For example, you can assign one address (area1 plus system ID), and two additional addresses in different areas (area2 plus system ID and area3 plus system ID) where the system ID is the same.

A router can dynamically learn about any adjacent router. As part o this process, the routers inform each other of their area addresses. If two routers share at least one area address, the set of area addresses of the two routers are merged. The merged set cannot contain more than three addresses. If there are more than three, the three addresses with the lowest numerical values are kept, and all others are dropped.

Beginning in global configuration mode, configure multiple area ad dresses in IS-IS areas:

1. Enable IS-IS routing and enter router configuration mode.

   router isis [*tag*]

2. Configure NETs for the routing process. The router can have up to three NETs. Enter each command separately.

Examples: NETs Configuration

These are examples of configuring NETs for both ISO IGRP and IS-IS.

ISO IGRP

This example illustrates specifying a NET:

```
router iso-igrp Finance
net 47.0004.004d.0001.0000.0c11.1111.00
```

This example illustrates using a name for a NET:

```
clns host NAME 39.0001.0000.0c00.1111.00
router iso-igrp Marketing
net NAME
```

The use of this **`net`** router configuration command configures the system ID, area address, and d omainaddress. Only a single NET per routing process is allowed.

```
router iso-igrp local
net 49.0001.0000.0c00.1111.00
```

IS-IS

This example illustrates specifying a single NET:

```
router isis Pieinthesky
net 47.0004.004d.0001.0000.0c11.1111.00
```

This example illustrates using a name for a NET:

```
clns host NAME 39.0001.0000.0c00.1111.00
router isis
net NAME
```

IS-IS Multihoming

This example illustrates the assignment of three separate area ad dresses for a single router using net commands. Traffic received that includes an area address of 47.0004.004d.0001, 47.0004.004d.0002, or 47.0004.004d.0003, and that has the same system ID, is forwarded to this router.

```
router isis eng-area1
! |       IS-IS Area|    System ID| S|
net 47.0004.004d.0001.0000.0C00.1111.00
net 47.0004.004d.0002.0000.0C00.1111.00
net 47.0004.004d.0003.0000.0C00.1111.00
```

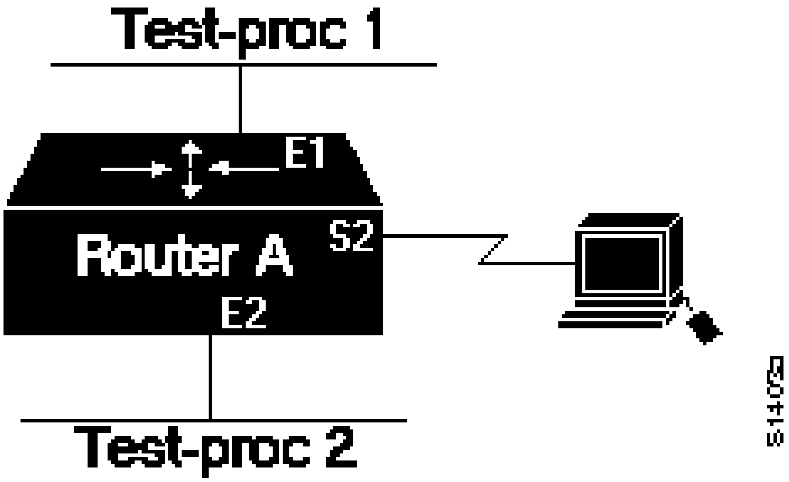
Example: Router in Two Areas

These two examples show how to configure a router in two areas. The first example configures ISO IGRP; the second configures IS-IS.

ISO IGRP

In this example, the router is in domai n49.0001 and has a system ID of aaaa.aaaa.aaaa. The router is in two areas: 31 and 40 (decimal). Figure 9-8 illustrates this configuration.

Figure 9-8:ISO IGRP Configuration



```
router iso-igrp test-proc1
```

001F in this example net is the hex value for area 31:

```
net 49.0001.001F.aaaa.aaaa.aaaa.00
router iso-igrp test-proc2
```

0028 in this example net is the hex value for area 40:

```
net 49.0001.0028.aaaa.aaaa.aaaa.00
!
interface ethernet 1
clns router iso-igrp test-proc1
!
interface serial 2
clns router iso-igrp test-proc1
!
interface ethernet 2
clns router iso-igrp test-proc2
```

IS-IS

To run IS-IS instead of ISO IGRP, use this configuration. The illustration in Figure 9- still applies. Ethernet interface 2 is configured for IS-IS routing and is assigned the tag of test-proc2.

```
router iso-igrp test-proc1
net 49.0002.0002.bbbb.bbbb.bbbb.00
router isis test-proc2
net 49.0001.0002.aaaa.aaaa.aaaa.00
!
interface ethernet 1
clns router iso-igrp test-proc1
!
interface serial 2
clns router iso-igrp test-proc1
!
interface ethernet 2
clns router IS-IS test-proc2
```

To allow CLNS packets only to blindly pass through an interface without routing updates, you could use a simple configuration. This example shows such a configuration:

```
clns routing
interface serial 2
```

This permits serial 2 to pass CLNS packets without having CLNS routing turned on:

```
clns enable
```

Configure IS-IS Parameters

Cisco's IS-IS implementation allows you to customize certain IS-IS parameters. You can perform the optional tasks:

- Specify Router-Level S upport
- Configure IS-IS Authentication Passwords
- Ignore IS-IS Link-State Packet (LSP) Errors
- Log Adjacency State Changes
- Change IS-IS LSP MTU Size

Specify Router-Lev elSupport

It is seldom necessary to configure the IS type because the IS-IS protocol will automatically establish this. However, you can configure th AI2524 to act as a Le vel1 (intra-area) router, as both a Level1 router and a Leve l2 (interarea) router, or as an interarea router only.

In router configuration mode, configure the IS-IS level at which the router is to operate:

```
is-type {level-1 | level-1-2 | level-2-
only}
```

Configure IS-IS Authentication Passwords

You can assign authentication passwords to areas and domains. An area password is inserted in Level 1 (station router) link-state PDUs (LSPs), complete sequence number PDUs (CSNPs), and partial sequence number PDUs (PSNPs). A routi ng domainauthentication password is inserted in Leve l2 (area router) LSP, CSNP, and PSNP.

In router configuration mode, configure area or domain passwords:

1. Configure the area authentication password.

   area-password ***password***

2. Configure the routin g domainauthentication password.

   domain-password ***password***

Ignore IS-IS Link-State Packet (LSP) Errors

You can configure the router to ignore IS-IS LSPs that are received with internal checksum errors, rather than purging the LSPs. LSPs are used by the receiving routers to maintain their routing tables.

The IS-IS protocol definition requires that a received LSP with an incorrect data-link checksum be purged by the receiver, which causes the initiator of the LSP to regenerate it. However, if a network has a link that causes data corruption while still delivering LSPs with correct data-link checksums, a continuous cycle of purging and regenerating large numbers of LSPs can occur, rendering the network nonfunctional.

In router configuration mode, allow the router to ignore LSPs with an internal checksum error:

```
ignore-lsp-errors
```

Log Adjacency State Changes

You can configure IS-IS to generate a log message when an IS-IS ad jacency changes state (up or down). This may be useful when monitoring large networks. Messages are logged using the system error message facility. Messages are in this form:

```
%CLNS-5-ADJCHANGE: ISIS: Adjacency to 0000.0000.0034 (Serial0) Up, new adjacency
%CLNS-5-ADJCHANGE: ISIS: Adjacency to 0000.0000.0034 (Serial0) Down, hold time
expired
```

In router configuration mode, generate log messages when an IS-IS adjacency changes state:

`log-adjacency-changes`

Change IS-IS LSP MTU Size

Under normal conditions, the default maximum transmission unit (MTU) size should be sufficient. However, if the MTU of a link is lowered to less than 1500 bytes, the LSP MTU must be lowered accordingly on each router in the network. If this is not done, routing will become unpredictable.

The MTU size must be less than or equal to the smallest MTU of any link in the network. The default size is 1497 bytes.

Caution: ***The CLNS MTU of a link (which is the applicable value for IS-IS, even if it is being used to route IP) may differ from the IP MTU. To be certain about a link MTU as it pertains to IS-IS, use the* `show clns` *interface command to display the value.***

In router configuration mode, change the MTU size of IS-IS link state packets:

`lsp-mtu` ***`size`***

Warning: ***This rule applies for all routers in a network. If any link in the network has a reduced MTU, all routers must b changed, not just the routers directly connected to th link.***

Configure IS-IS Interface Parameters

Cisco's IS-IS implementation allows you to customize certain interface-specific IS-IS parameters:

- Adjust IS-IS Link-State Metrics
- Set the Advertised Hello Interval and Hello Multiplier
- Set the Advertised CSNP Interval
- Set the Retransmission Interval
- Specify Designated Router Election
- Specify the Interface Circuit Type
- Configure IS-IS Authentication Passwords

You are not required to alter any of these parameters, but some inter face parameters must be consistent across all routers in the network. Therefore, if you do configure any of these parameters, be sure the configurations for all routers on the network have compatible values.

Adjust IS-IS Link-State Metrics

You can configure a cost for a specified interface. The default metric is used as a value for the IS-IS metric. This is the value assigned when there is no quality of service (QOS) routing performed. The only metric that is supported by the Cisco IOS software and that you can configure is the default-metric, which you can configure for L evel1 or Level 2 routing or both.

In interface configuration mode, configure the link state metric:

```
isis metric default-metric {level-1 |
level-2}
```

Set the Advertised Hello Interval and Hello Multiplier

You can specify the length of time (in seconds) between hello packets that the Cisco IOS software sends on the interface. You can also change the default hello packet multiplier used on the interface to determine the hold time transmitted in IS-IS hello packets (the default is 3).

The hold time determines how long a neighbor waits for another hello packet before declaring the neighbor down. This time determines how quickly a failed link or neighbor is detected so that routes can be recalculated.

In interface configuration mode, set the advertised hello interval and multiplier:

1. Specify the length of time, in seconds, between hello packets th software sends on the specified interface.

   ```
   isis hello-interval seconds {level-1 |
   level-2}
   ```

2. Specify the number used to multiply the hello interval seconds by to determine the total holding time transmitted in the IS-IS hello packet. If not specified, a multiplier of 3 is used.

   ```
   isis hello-multiplier multiplier [{level-1
   | level-2}]
   ```

The hello interval can be configured independently for Level1 and Level 2, except on serial point-to-point interfaces. (Because there is only a single type of hello packet sent on serial links, the hello packet is independent of Lev el1 or Lev el2.) Specify an optional level for X.25, SMDS, and Frame Relay multiaccess networks.

Use the **isis hello-multiplier** command in circumstances where hello packets are lost frequently and IS-IS adjacencies are failing unnecessarily. You can raise the hello multiplier and lower th hello interval (**isis hello-interval** command) correspond ingly to make the hello protocol more reliable without increasing th time required to detect a link failure.

Set the Advertised CSNP Interval

CSNPs are sent by the designated router to maintain database synchronization.

In interface configuration mode, configure the IS-IS CSNP interval fo the interface:

```
isis csnp-interval seconds {level-1 |
level-2}
```

This feature does not apply to serial point-to-point interfaces. It does apply to WAN connections if the WAN is viewed as a multiaccess meshed network.

Set the Retransmission Interval

In interface configuration mode, configure the number of seconds between retransmission of LSPs for point-to-point links.

```
isis retransmit-interval  seconds
```

The value you specify should be an integer greater than the expected round-trip delay between any two routers on the network. The setting of this parameter should be conservative, or needless retransmission will result. The value you determine should be larger for serial lines and virtual links.

Specify Designated Router Election

You can configure the priority to use for designated router election. Priorities can be configured for Level1 an d Level2 individually. Th designated router enables a reduction in the number of adjacencies required on a multiaccess network, which in turn reduces the amount of routing protocol traffic and the size of the topology database.

In interface configuration mode, configure the priority to use for designated router election:

```
isis priority  value {level-1 | level-2}
```

Specify the Interface Circuit Type

It is normally not necessary to configure this feature because the IS-IS protocol automatically determines area boundaries and keeps L evel1 and Leve l2 routing separate. However, you can specify the adjacency levels on a specified interface.

In interface configuration mode, configure the adjacency for neighbors on the specified interface:

```
isis circuit-type {level-1 | level-1-2 |
level-2-only}
```

If you specify Level 1, a Level 1 adjacency is established if there is at least one area address common to both this node and its neighbors.

If you specify both Le vel1 and L evel2 (the default value), a Level1 and 2 adjacency is established if the neighbor is also configured as both Leve l1 and Lev el2 and there is at least one area in common. If there is no area in common, a Leve l2 adjacency is established.

If you specify Leve l2 only, a Le vel2 adjacency is established. If the neighbor router is a Leve l1 router, no adjacency is established.

Configure IS-IS Password Authentication

You can assign different authentication passwords for different routing levels. By default, authentication is disabled. Specifying Lev el1 or Level 2 enables the password only for Le vel1 or L evel2 routing, respectively. If you do not specify a level, the default is Le vel1.

In interface configuration mode, configure an authentication password for an interface:

```
isis password password {level-1 | level-2}
```

Configure CLNS Static Routing

You do not need to explicitly specify a routing process to use stati routing facilities. You can enter a specific static route and apply it globally, even if you have configured the router for ISO IGRP or IS-IS dynamic routing.

To configure a static route, complete the tasks in the following sections. Only enabling static routes is required; the remaining tasks may be necessary for certain applications, but are otherwise optional.

- Enable Static Routes
- Configure Variations of the Static Route
- Map NSAP Addresses to Media Addresses

Enable Static Routes

To configure static routing, you must enable CLNS on the router and on the interface. CLNS routing is enabled on the router by default when you configure ISO IGRP or IS-IS routing protocols. NSAP addresses that start with the NSAP prefix you specify are forwarded to the next-hop node.

In global configuration mode, configure CLNS on the router:

1. Configure CLNS.

   ```
   clns routing
   ```

2. Assign an NSAP address to the router if the router has not been configured to route CLNS packets dynamically using ISO IGRP or IS-IS.

   ```
   clns net {net-address | name}
   ```

3. Enter a specific static route.

   ```
   clns route nsap-prefix {next-hop-net |
   name}
   ```

Warning:** **If you have not configured the router to route CLNS packets dynamically using ISO IGRP or IS-IS, you must as sign an address to the router.

You also must enable ISO CLNS for each interface you want to pass ISO CLNS packet traffic to end systems, but for which you do not want to perform any dynamic routing on the interface. This is done automatically when you configure IS-IS or ISO IGRP routing on an interface; however, if you do not intend to perform any dynamic routing on an interface, you must manually enable CLNS. You can assign an NSAP address for a specific interface. This allows the Cisco IOS software to advertise different addresses on each interface. This is useful if you are doing static routing and need to control the source NET used by the router on each interface.

In interface configuration mode, configure CLNS on an interface:

1. Enable ISO CLNS for each interface.

   `clns enable`

2. Optionally, assign an NSAP address to a specific interface.

   clns net {*nsap-address* | *name*}

Examples: Basic Static Routing

Configuring FDDI, Ethernets, Token Rings, and serial lines for CLNS can be as simple as enabling CLNS on the interfaces. This is all that is ever required on serial lines using HDLC encapsulation. If all systems on an Ethernet or Token Ring support ISO 9542 ES-IS, then nothing else is required.

Example 1

This example illustrates how an Ethernet and a serial line can be configured:

1. Enable clns packets to be routed:

```
clns routing
```

2. Configure this network entity title for the routing process:

```
clns net 47.0004.004d.0055.0000.0C00.BF3B.00
```

3. Pass ISO CLNS traffic on ethernet 0 to end systems without routing:

```
interface ethernet 0
clns enable
```

4. Pass ISO CLNS traffic on serial 0 to end systems without routing:

```
interface serial 0
clns enable
```

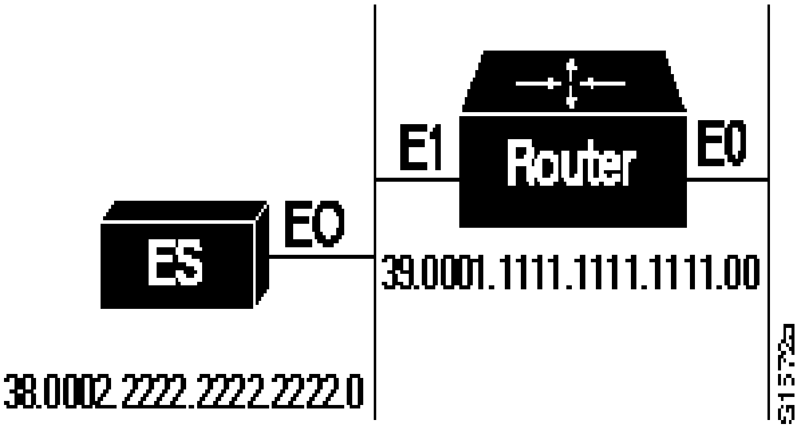
5. Create a static route for the interface:

```
clns route 47.0004.004d.0099 serial 0
clns route 47.0005 serial 0
```

Example 2

This is a more complete example of CLNS static routing on a system with two Ethernet interfaces. After configuring routing, you define NET and enable CLNS on the Ethernet 0 and Ethernet 1 interfaces. You must then define an ES neighbor and define a static route with the **`clns route`** global configuration command, as shown. In this situation, there is an ES on Ethernet 1 that does not support ES-IS.Figur 9-9 illustrates this network.

Figure 9-9:Static Routing



```
clns host sid 39.0001.1111.1111.1111.00
clns host bar 39.0002.2222.2222.2222.00
```

1. Assign a static address for the router:

```
clns net sid
```

2. Enable CLNS packets to be routed:

```
clns routing
```

3. Pass ISO CLNS packet traffic to end systems without routing them:

```
interface ethernet 0
clns enable
```

4. Pass ISO CLNS packet traffic to end systems without routing them:

```
interface ethernet 1
clns enable
```

5. Specify end system for static routing:

```
clns es-neighbor bar 0000.0C00.62e7
```

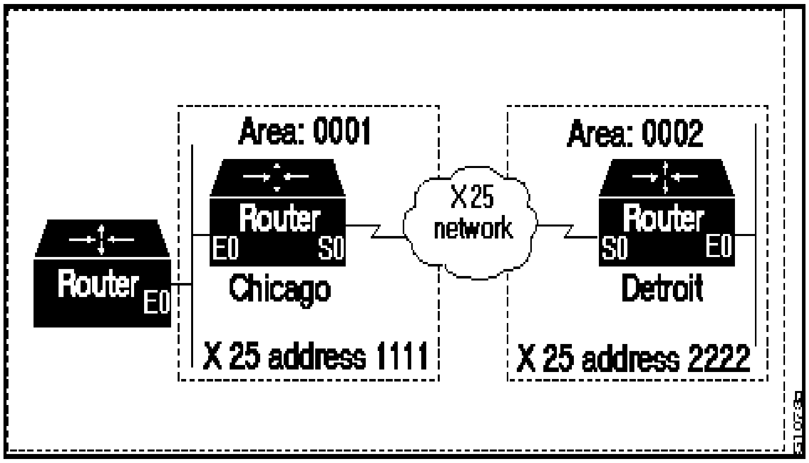
6. Create an interface-static route to bar for packets with the this NSAP address:

```
clns route 47.0004.000c bar
```

Example: Static Intradomain Routing

Figure 9-10, and the configurations demonstrate how to use static routing inside of a domain. Imagine a company with branch offices in Detroit and Chicago, connected with an X.25 link. These offices are both in the domai nnamed Sales.

Figure 9-10:CLNS X.25 Intradomain Routing



This example shows one way to configure the router in Chicago:

1. Define the name chicago to be used in place of this NSAP:

```
clns host chicago 47.0004.0050.0001.0000.0c00.243b.00
```

2. Define the name detroit to be used in place of this NSAP:

```
clns host detroit 47.0004.0050.0002.0000.0c00.1e12.00
```

3. Enable ISO IGRP routing of CLNS packets:

```
router iso-igrp sales
```

4. Configure net chicago, as shown in steps 1-3:

```
net chicago
```

5. Specify iso-igrp routing using the specified tag sales:

```
interface ethernet 0
clns router iso-igrp sales
```

6. Set the interface up as a DTE with X.25 encapsulation:

```
interface serial 0
encapsulation x25
x25 address 1111
x25 nvc 4
```

7. Specify iso-igrp routing using the specified tag sales:

```
clns router iso-igrp sales
```

8. Define a static mapping between Detroit's nsap and its X.121 address:

```
x25 map clns 2222 broadcast
```

This configuration brings up an X.25 virtual circuit between the router in Chicago and the router in Detroit. Routing updates will be sent across this link. This implies that the virtual circuit could be up continuously.

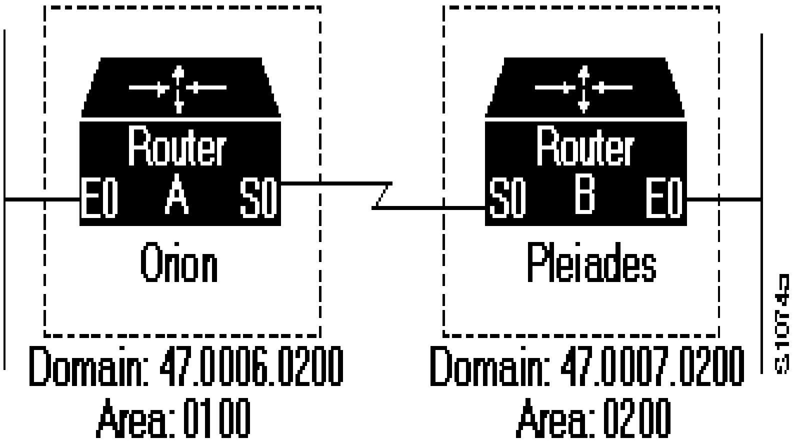
If the Chicago office should grow to contain multiple routers, it would be appropriate for each of those routers to know how to get to Detroit. Add the following command to redistribute information between routers in Chicago:

```
router iso-igrp sales
redistribute static
```

Example: Static InterdomainRo uting

Figure 9-11 and the example configurations illustrate how to configure two routers that distribute information across domains. In this example, Router A (in dom ainOrion) and Router B (i n domainPleiades) communicate across a serial link.

Figure 9-11:CLNS Interdomain S tatic Routing



Router A

This configuration shows how to configure Router A for stati interdomainrouting:

1. Define tag orion for net 47.0006.0200.0100.0102.0304.0506.00:

```
router iso-igrp orion
```

2. Configure this network entity title for the routing process:

```
net 47.0006.0200.0100.0102.0304.0506.00
```

3. Define the tag bar to be used in place of Router B's NSAP:

```
clns host bar 47.0007.0200.0200.1112.1314.1516.00
```

4. Specify iso-igrp routing using the specified tag orion:

```
interface ethernet 0
clns router iso-igrp orion
```

5. Pass ISO CLNS traffic to end systems without routing:

```
interface serial 1
clns enable
```

6. Configure a static route to Router B:

```
clns route 39.0001 bar
```

Router B

This configuration shows how to configure Router B for static interdomainrouting:

```
router iso-igrp pleiades
```

1. Configure the network entity title for the routing process:

```
net 47.0007.0200.0200.1112.1314.1516.00
```

2. Define the name sid to be used in place of Router A's NSAP:

```
clns host sid 47.0006.0200.0100.0001.0102.0304.0506.00
```

3. Specify iso-igrp routing using the specified tag pleiades:

```
interface ethernet 0
clns router iso-igrp pleiades
```

4. Pass ISO CLNS traffic to end systems without routing:

```
interface serial 0 clns enable
```

5. Pass packets bound for sid in dom ain47.0006.0200 through serial 0:

```
clns route 47.0006.0200 sid
```

CLNS routing updates will not be sent on the serial link; however, CLNS packets will be sent and received over the serial link.

Configure Variations of the Static Route

These tasks include variations of the **clns route** global configuration command:

- Bind the next hop to a specified interface and media address when you do not know the NSAP address of your neighbor. Note that this version of the clns route command is not literally applied to a specific interface.

  clns route ***nsap-prefix type number*** **[*snpa-address*]**

- Tell the Cisco IOS software to discard packets with the specified NSAP prefix.

  clns route ***nsap-prefix*** **discard**

- Specify a default prefix.

  clns route default ***nsap-prefix type number***

Map NSAP Addresses to Media Addresses

Conceptually, each ES lives in one area. It discovers the nearest IS by listening to ES-IS packets. Each ES must be able to communicate di rectly with an IS in its area.

When an ES wants to communicate with another ES, it sends the packet to any IS on the same medium. The IS looks up the destination NSAP address and forwards the packet along the best route. If the destination NSAP address is for an ES in another area, the Level1 IS sends the packet to the nearest Lev el2 IS. The Lev el2 IS forwards the packet along the best path for the destination area until it gets to a Level 2 IS that is in the destination area. This IS then forwards the packet along the best path inside the area until it is delivered to the destination ES.

ESs need to know how to get to a Level1 IS for their area, a nd Level1 ISs need to know all of the ESs that are directly reachable through each of their interfaces. To provide this information, the routers support the

ES-IS protocol. The router dynamically discovers all ESs running the ES-IS protocol. ESs that are not running the ES-IS protocol must be configured statically.

It is sometimes desirable for a router to have a neighbor configured statically rather than learned through ES-IS, ISO IGRP, or IS-IS.

Warning:** **It is necessary to use static mapping only for ESs that do not support ES-IS. The Cisco IOS software continues to dynamically discover ESs that do support ES-IS. If you have configured interfaces for ISO IGRP or IS-IS, the ES-IS routing software automatically turns on ES-IS for those interfaces.

In interface configuration mode, enter static mapping information between the NSAP protocol addresses and the subnetwork point of at tachment (SNPA) addresses (media) for ESs or ISs, as needed:

- Configure all end systems that will be used when you manually specify the NSAP-to-SNPA mapping.

  ```
  clns es-neighbor nsap snpa
  ```

- Configure all intermediate systems that will be used when you manually specify the NSAP-to-SNPA mapping.

  ```
  clns is-neighbor nsap snpa
  ```

For more information, refer to Configure CLNS over WANs.

Warning: The SNPA is a data link layer address (such as an Ethernet address, X.25 address, or Frame Relay DLCI address) used to configure a CLNS route for an interface.

Configure Miscellaneous Features

Perform these optional tasks to configure miscellaneous features of an ISO CLNS network:

- Specify Shortcut NSAP Addresses
- Use the IP Domain Name System to Discover ISO CLNS Ad dresses
- Create Packet-Forwarding Filters and Establish Adjacencies
- Redistribute Routing Information
- Specify Preferred Routes
- Configure ES-IS Hello Packet Parameters

Specify Shortcut NSAP Addresses

You can define a name-to-NSAP address mapping. This name can then be used in place of typing the long set of numbers associated with an NSAP address.

In global configuration mode, define a name-to-NSAP address mapping:

```
clns host name nsap
```

The assigned NSAP name is displayed, where applicable, in **show** and **debug** EXEC commands. However, some effects and requirements are associated with using names to represent NETs and NSAP ad dresses.

The **clns host** global configuration command is generated after all other CLNS commands when the configuration file is parsed. As a result, you cannot edit the nonvolatile random access memory (NVRAM) version of the configuration to specifically change the address defined in the original **clns host** command. You must specifically change any commands that refer to the original address. This affects all commands that accept names.

The commands that are affected by these requirements include:

- **net** (router configuration command
- **clns is-neighbor** (interface configuration command)
- **clns es-neighbor** (interface configuration command)
- **clns route** (global configuration command)

Use the IP DomainNa me System to Discover ISO CLNS Addresses

If your router has both ISO CLNS and IP enabled, you can use the Domain Naming System (DNS) to query ISO CLNS addresses by using the NSAP address type, as documented in RFC 1348. This feature is useful for the ISO CLNS **ping** EXEC command and when making Telnet connections. This feature is enabled by default.

In global configuration mode, enable or disable DNS queries for IS CLNS addresses:

- Enable DNS queries for CLNS addresses.

  ip domain-lookup nsap

- Disable DNS queries for CLNS addresses.

  no ip domain-lookup nsap

Create Packet-Forwarding Filters and Establish Adjacencies

You can build powerful CLNS filter expressions, or access lists. These can be used to control either the forwarding of frames through route interfaces, or the establishment of adjacencies with, or the application of filters to, any combination of ES or IS neighbors, ISO IGRP neighbors, or IS-IS neighbors.

CLNS filter expressions are complex logical combinations of CLNS filter sets. CLNS filter sets are lists of address templates against which CLNS addresses are matched. Address templates are CLNS address patterns that are either simple CLNS addresses that match just one address, or match multiple CLNS addresses through the use of wildcard characters, prefixes, and suffixes. Frequently used address templates can be given aliases for easier reference.

1. In global configuration mode, establish CLNS filters:

- Create aliases for frequently used address templates.

  clns template-alias *name template*

- Build filter sets of multiple address template permit and deny conditions.

  clns filter-set *sname* [permit | deny] template

- Build filter expressions, using one or more filter sets.

  clns filter-expr *ename term*

2. In interface configuration mode, apply filter expressions to an interface:

- Apply a filter expression to frames forwarded in or out of an interface.

  clns access-group ***name*** **[in | out]**

- Apply a filter expression to ISIS adjacencies.

  isis adjacency-filter ***name*** **[match-all]**

- Apply a filter expression to ISO IGRP adjacencies.

  iso-igrp adjacency-filter ***name***

- Apply a filter expression to ES or IS adjacencies.

  clns adjacency-filter {es | is} name

Examples: CLNS Filter

This example allows packets if the address starts with either 47.0005 or 47.0023. It implicitly denies any other address.

```
clns filter-set US-OR-NORDUNET permit 47.0005...
clns filter-set US-OR-NORDUNET permit 47.0023...
```

Redistribute Routing Information

In addition to running multiple routing protocols simultaneously, the Cisco IOS software can redistribute information from one routing process to another.

You can also configure the Cisco IOS software to do interdomain d ynamic routing by configuring two routing processes and two NETs (thereby putting the router into two domains) and redistributing the routing information between the domains. Routers configured this way are referred to as border routers. If you have a router that is in two routing domains, you might want to redistribute routing information between the two domains.

Warning: It is not necessary to use redistribution between areas. Redistribution only occurs for Le vel2 routing.

1. In global configuration mode, configure the router to redistribute routing information into the ISO IGRP domain:

- Specify the routing protocol and tag (if applicable) into which you want to distribute routing information.

  router iso-igrp [***tag*****]**

- Specify one or more ISO IGRP routing protocol and tag (if applicable) you want to redistribute.

  redistribute iso-igrp [***tag*****] [route-map** ***map-tag*****]**

- Specify the IS-IS routing protocol and tag (if applicable) you want to redistribute.

  redistribute isis [***tag*****] [route-map** ***map-tag*****]**

- Specify the static routes you want to redistribute.

  redistribute static [clns | ip]

2. Beginning in global configuration mode, configure the router to redistribute routing information into the IS-IS domains:

- Specify the routing protocol and tag (if applicable) into which you want to distribute routing information.

  router isis [***tag*****]**

- Specify the IS-IS routing protocol and tag (if applicable) you want to redistribute.

  redistribute isis [***tag*****] [route-map** ***map-tag*****]**

Warning: By default, static routes are redistributed into IS-IS.

You can conditionally control the redistribution of routes between routing domains by defining route maps between the two domains. Route maps allow you to use tags in routes to influence route redistribution.

3. In global configuration mode, conditionally control the redistribution of routes between domains:

   route-map ***map-tag*** **{permit | deny}** ***sequence-number***

One or more **match** command and one or more **set** commands typically follow a **route-map** command to define the conditions for redistributing routes from one routing protocol into another. If there are no **match** commands, everything matches. If there are no **set** commands, nothing is done (other than the match).

Each **route-map** command has a list of **match** and **set** commands associated with it. The **match** commands specify the match criteria—the conditions under which redistribution is allowed for the current

route-map command. The **set** commands specify the redistribution set actions—the particular redistribution actions to perform if th criteria enforced by th **match** commands are met. When all matc criteria are met, all **set** actions are performed

The **match route-map** configuration command has multiple formats. The **match** commands may be given in any order, and all defined match criteria must be satisfied to cause the route to be redistributed according to th **set** actions given with the **set** commands.

In route-map configuration mode, define the match criteria for redistribution of routes from one routing protocol into another by performing at least one of these

- Match routes that have a network address matching one or more of the specified names (the names can be a standard access list, filter set, or expression).

  match clns address ***name*** **[*name...name*]**

- Match routes that have a next hop address matching one or more of the specified names (the names can be a standard access list, filter set, or expression).

  match clns next-hop ***name*** **[*name...name*]**

- Match routes that have been advertised by routers matching one or more of the specified names (the names can be a standard access list, filter set, or expression).

  match clns route-source ***name*** **[*name...name*]**

- Match routes that have the next hop out matching one or more o the specified interfaces.

  match clns interface ***type number*** **[*type number...type number*]**

- Match routes that have the specified metric.

  match metric ***metric-value***

- Match routes that have the specified route type.

  match route-type {level-1 | level-2}

In route-map configuration mode, define **set** actions for redistribution of routes from one routing protocol into another by performing at least one of these:

- Set the routing level of the routes to be advertised into a specified are of the routing domain.

  set level {level-1 | level-2 | level-1-2}

- Set the metric value to give the redistributed routes.

  set metric *metric-value*

- Set the metric type to give the redistributed routes.

  set metric-type {internal | external}

- Set the tag value to associate with the redistributed routes.

  set tag *tag-value*

Examples: Route Map

This example redistributes two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first routes are OSPF external IP routes with tag 5, and these are inserted into level-2 IS-IS LSPs with a metric of 5. The second routes are ISO IGRP derived CLNS prefix routes that match CLNS filter expression osifilter. These are redistributed into IS-IS as level-2 LSPs with a metric of 30.

```
router isis
redistribute ospf 109 route-map ipmap
redistribute iso-igrp nsfnet route-map osimap
!
route-map ipmap permit
match route-type external
match tag 5
set metric 5
set level level-2
!
route-map osimap permit
match clns address osifilter
set metric 30
clns filter-set osifilter permit 47.0005.80FF.FF00
```

Given this configuration, a RIP learned route for network 160.89.0.0 and an ISO IGRP learned route with prefix 49.0001.0002 will be redistributed into an IS-IS level-2 LSP with a metric of 5. For example:

```
router isis
redistribute rip route-map ourmap
redistribute iso-igrp remote route-map ourmap
!
route-map ourmap permit
 match ip address 1
match clns address ourprefix
set metric 5
set level level-2
!
access-list 1 permit 160.89.0.0 0.0.255.255
clns filter-set ourprefix permit 49.0001.0002...
```

Specify Preferred Routes

When multiple routing processes are running in the same router for CLNS, it is possible for the same route to be advertised by more than one routing process. The Cisco IOS software always picks the route whose routing protocol has the lowest administrative distance. The lower the value of the distance, the more preferred the route.

By default, these administrative distances are assigned:

- Static routes—10
- ISO IGRP routes—100
- IS-IS routes—110

In router configuration mode, change an administrative distance for a route:

distance *value* [clns]

If you want an ISO IGRP prefix route to override a static route, you must set the distance for the routing process to be lower than 10.

Configure ES-IS Hello Packet Parameters

You can configure ES-IS parameters for communication between end systems and routers. In general, you should leave these parameters at their default values.

When configuring an ES-IS router, be aware of these items:

- ES-IS does not run over X.25 links unless the broadcast facility is enabled.

- ES hello packets and IS hello packets are sent without options. Options in received packets are ignored.

ISs and ESs periodically send out hello packets to advertise their availability. The frequency of these hello packets can be configured.

The recipient of a hello packet creates an adjacency entry for the system that sent it. If the next hello packet is not received within the interval specified, the adjacency times out and the adjacent node is considered unreachable. A default rate has been set for hello packets and packet validity, but it can be changed.

In global configuration mode, change the defaults:

- Specify the rate at which ES hello and IS hello packets are sent.

  `clns configuration-time` ***seconds***

- Allow the sender of an ES hello or IS hello packet to specify the length of time you consider the information in these packets to be valid.

  `clns holding-time` ***seconds***

A default rate has been set for the ES Configuration Timer (ESCT) option, but it can be changed.

In interface configuration mode, specify how often the end system should transmit ES hello packet PDUs:

`clns esct-time` ***seconds***

Configure CLNS over WANs

This section provides general information about running ISO CLNS over WANs.

You can use CLNS routing on serial interfaces with HDLC, PPP, LAPB, X.25, Frame Relay, DDR, or SMDS encapsulation. To use HDLC encapsulation, you must have a router at both ends of the link. If you use X.25 encapsulation, and if IS-IS or ISO IGRP is not used on an interface, you must manually enter the NSAP-to-X.121 address mapping. The LAPB, SMDS, Frame Relay, and X.25 encapsulations interoperate with other vendors.

Both ISO IGRP and IS-IS can be configured over WANs.

X.25 is not a broadcast medium and, therefore, does not broadcast protocols (such as ES-IS) that automatically advertise and record mappings between NSAP/NET (protocol addresses) and SNPA (media addresses). (With X.25, the SNPAs are the X.25 network addresses, or the X.121 addresses. These are usually assigned by the X.25 network provider.) If you use static routing, you must configure the NSAP-to X.121 address mapping with th **`x25 map`** command.

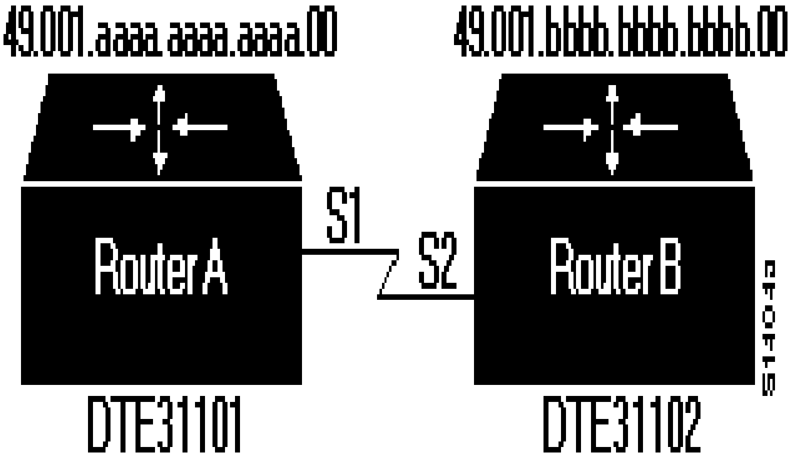
Configuring a serial line to use CLNS over X.25 requires configuring the general X.25 information and the CLNS-specific information. First, configure the general X.25 information. Then, enter the CLNS static mapping information.

You can specify X.25 nondefault packet and window sizes, reverse charge information, and so on. The X.25 facilities information that can be specified is exactly the same as in the **`x25 map`** interface configuration command.

Example: ISO CLNS over X.25

In this example, serial interface 1 on Router A acts as a DTE for X.25. It permits broadcasts to pass through. Router B is an IS, which has CLNS address of 49.0001.bbbb.bbbb.bbbb.00 and an X.121 address of 31102. Router A has a CLNS address of 49.0001.aaaa.aaaa.aaaa.00 and an address of 31101. Figure 9-12 illustrates this configuration.

Figure 9-12:Routers Acting as DTEs and DCEs



Router A

```
router iso-igrp test-proc
net 49.0001.aaaa.aaaa.aaaa.00
!
interface serial 1
clns router iso-igrp test-proc
```

1. Assume the host is a DTE and encapsulates x.25:

```
encapsulation x25
```

2. Define the X.121 address of 31101 for serial 1:

```
X25 address 31101
```

3. Set up an entry for the other side of the X.25 link (Router B):

```
x25 map clns 31101 broadcast
```

Router B

```
router iso-igrp test-proc
net 49.0001.bbbb.bbbb.bbbb.00
!
interface serial 2
clns router iso-igrp test-proc
```

1. Configure this side as a DCE:

```
encapsulation x25-dce
```

2. Define the X.121 address of 31102 for serial 2:

```
X25 address 31102
```

3. Configure the NSAP of Router A and accept reverse charges:

```
x25 map clns 31101 broadcast accept-reverse
```

Enhance ISO CLNS Performance

Generally, you do not need to change the router's default settings for CLNS packet switching, but there are some modifications you can make when you decide to make changes in your network's performance. The following sections describe ISO CLNS parameters that you can change:

- Specify the MTU Size
- Disable Checksums
- Disable Fast Switching Through the Cache
- Set the Congestion Threshold
- Transmit Error Protocol Data Units (ERPDUs)
- Control Redirect Protocol Data Units (RDPDUs)
- Configure Parameters for Locally Sourced Packets

Specify the MTU Size

All interfaces have a default maximum packet size. However to reduc fragmentation, you can set the MTU size of the packets sent on the interface. The minimum value is 512; the default and maximum packet size depends on the interface type.

Changing the MTU value with the **`mtu`** interface configuration command can affect the CLNS MTU value. If the CLNS MTU is at its maximum given the interface MTU, the CLNS MTU will change with the interface MTU. However, the reverse is not true; changing th CLNS MTU value has no effect on the value for the **`mtu`** interface configuration command.

In interface configuration mode, set the CLNS MTU packet size for a specified interface:

```
clns mtu size
```

*Warning: **The CTR card does not support the switching of frames larger than 4472 bytes. Interoperability problems might occur if CTR cards are intermixed with other Token Ring cards on the same network. These problems can be mini mized by lowering the CLNS MTU sizes to be the same on all routers on the network.***

Disable Checksums

When the ISO CLNS routing software originates a CLNS packet, by default it generates checksums. In interface configuration mode, disable checksum generation:

```
no clns checksum
```

*Warning: **Enabling checksum generation has no effect on routing packets (ES-IS, ISO IGRP, and IS-IS) originated by th router; it applies to pings and traceroute packets.***

Disable Fast Switching Through the Cache

Fast switching through the cache is enabled by default for all supported interfaces. In interface configuration mode, disable fast switching:

```
no clns route-cache
```

*Warning: **The cache still exists and is used after the*** **`no clns route-cache`** ***interface configuration command is used; the software just does not do fast switching through the cache.***

Set the Congestion Threshold

If a router configured for CLNS experiences congestion, it sets the congestion-experienced bit. You can set the congestion threshold on a per-interface basis. By setting this threshold, you cause the system to set the congestion-experienced bit if the output queue has more than the specified number of packets in it.

In interface configuration mode, set the congestion threshold:

```
clns congestion-threshold number
```

Transmit Error Protocol Data Units (ERPDUs)

When a CLNS packet is received, the routing software looks in the routing table for the next hop. If it does not find one, the packet is discarded and an error protocol data unit (ERPDU) is sent.

You can set an interval between ERPDUs. Doing so reduces bandwidth if this feature is disabled. When you set the minimum interval between ERPDUs, the Cisco IOS software does not send ERPDUs more frequently than one per interface per ten milliseconds.

In interface configuration mode, transmit ERPDUs:

1. Send an ERPDU when the routing software detects an error in data PDU; this is enabled by default.

   ```
   clns send-erpdu
   ```

2. Set the minimum interval, in milliseconds, between ERPDUs.

   ```
   clns erpdu-interval milliseconds
   ```

Control Redirect Protocol Data Units (RDPDUs)

If a packet is sent out the same interface it came in on, a redirect protocol data unit (RDPDU) also can be sent to the sender of the packet. You can control RDPDUs with one of these actions:

- By default, CLNS sends RDPDUs when a better route for a given host is known. You can disable this feature. Disabling this feature reduces bandwidth because packets may continue to unnecessarily go through the router.

  ```
  clns send-rdpdu
  ```

- You can set the interval times between RDPDUs.

  ```
  clns rdpdu-interval milliseconds
  ```

Warning: SNPA masks are never sent, and RDPDUs are ignored by the Cisco IOS software when the router is acting as an IS.

Configure Parameters for Locally Sourced Packets

In global configuration mode, configure parameters for packets originated by a specified router:

- Specify in seconds the initial lifetime for locally generated packets.

  `clns packet-lifetime seconds`

- Specify whether to request ERPDUs on packets originated by the router.

  `clns want-erpdu`

You should set the packet lifetime low in an internetwork that has frequent loops.

Warning:* *The* `clns want-erpdu` *global configuration command has no effect on routing packets (ES-IS, ISO IGRP, and ISIS) originated by the router; it applies to pings and traceroute packets.

Example: Performance Parameters

This example shows how to set ES hello packet and IS hello packet parameters in a simple ISO IGRP configuration, as well as the MTU for a serial interface:

```
router iso-igrp xavier
net 49.0001.004d.0002.0000.0C00.0506.00
```

1. Send IS/ES hellos every 45 seconds:

```
clns configuration-time 45
```

2. Recipients of the hello packets keep information in the hellos for 2 minutes:

```
clns holding-time 120
```

3. Specify an mtu of 978 bytes; generally, do not alter the default:

```
mtu value interface serial 2
clns mtu 978
```

Monitor and Maintain the ISO CLNS Network

Use these EXEC commands to monitor and maintain the ISO CLNS caches, tables, and databases:

- Clear and reinitialize the CLNS routing cache.

  `clear clns cache`

- Remove ES neighbor information from the adjacency database.

  `clear clns es-neighbors`

- Remove IS neighbor information from the adjacency database.

  `clear clns is-neighbors`

- Remove CLNS neighbor information from the adjacency database.

  `clear clns neighbors`

- Remove dynamically derived CLNS routing information.

  `clear clns route`

- Invoke a diagnostic tool for testing connectivity.

  ping clns {*host* | *address*}

- Display information about the CLNS network.

  `show clns`

- Display the entries in the CLNS routing cache.

  `show clns cache`

- Display ES neighbor entries, including the associated areas.

  show clns es-neighbors [*type number*] [detail]

- Display filter expressions.

  show clns filter-expr [*name*] [detail]

- Display filter sets.

  show clns filter-set [*name*]

- List the CLNS-specific or ES-IS information about each interface.

  show clns interface [*type number*]

- Display IS neighbor entries, according to the area in which they are located.

  show clns is-neighbors [*type number*] [detail]

- Display both ES and IS neighbors.

  show clns neighbors [*type number*] [detail]

- List the protocol-specific information for each IS-IS or ISO IGRP routing process in this router.

  show clns protocol [*domain* | *area-tag*]

- Display all the destinations to which this router knows how to route packets.

  show clns route [*nsap*]

- Display information about the CLNS packets this router has seen.

  show clns traffic

- Display the IS-IS link state database.

  show isis database [level-1] [level-2] [l1] [l2] [detail] [*lspid*]

- Display the IS-IS Leve l1 routing table.

  show isis routes

- Display a history of the SPF calculations for IS-IS.

  show isis spf-log

- Display all route maps configured or only the one specified.

  show route-map [*map-name*]

- Discover the paths taken to a specified destination by packets in the network.

  trace clns *destination*

- Display the routing table in which the specified CLNS destination is found.

  which-route {*nsap-address* | *clns-name*}

Configure TARP on ISO CLNS

Some applications (typically used by telephone companies) running on Synchronous Optical Network (SONET) devices identify these devices by a target identifier (TID). Therefore, it is necessary for the router to cache TID-to-network address mappings. Because these ap plications usually run over OSI, the network addresses involved in the mapping are OSI NSAPs.

When a device must send a packet to another device it does not know about (that is, it does not have information about the NSAP address corresponding to the remote device's TID), the device needs a way to request this information directly from the device, or from an intermediate device in the network. This functionality is provided by an address resolution protocol called Target Identifier Address Resolution Protocol (TARP).

Requests for information and associated responses are sent as TARP protocol data units (PDUs), which are sent as CLNP data packets. TARP PDUs are distinguished by a unique N-selector in the NSAP address. Here are the five types of TARP PDUs:

- Type 1—Sent when a device has a TID for which it has no matching NSAP. Type 1 PDUs are sent to all Le vel1 (IS-IS and ES-IS) neighbors. If no response is received within the specified time limit, a Type 2 PDU is sent. To prevent packet looping, a loop detection buffer is maintained on the router. A Type 1 PDU is sent when you use the **tarp resolve** command.
- Type 2—Sent when a device has a TID for which it has no matching NSAP and no response was received from a Type 1 PDU. Type 2 PDUs are sent to all Le vel1 and Lev el2 neighbors. A time limit for Type 2 PDUs can also be specified. A Type 2 PDU is sent when you use the **tarp resolve** command and specify the option 2.
- Type 3—Sent as a response to a Type 1, Type 2, or Type 5 PDU. Type 3 PDUs are sent directly to the originator of the request.
- Type 4—Sent as a notification when a change occurs locally (fo example, a TID or NSAP change). A Type 4 PDU is sent when you use the **tarp query** command.
- Type 5—Sent when a device needs a TID that corresponds to a specific NSAP. Unlike Type 1 and Type 2 PDUs that are sent to all Leve l1 and Level 2 neighbors, a Type 5 PDU is sent only to a particular router.

In addition to the type, TARP PDUs contain the sender's NSAP, th sender's TID, and the target's TID (if the PDU is a Type 1 or Type 2).

TARP Configuration Task List

To configure TARP on the router, complete these tasks (only the first task is required, all other tasks are optional):

- Enable TARP and Configure a TARP TID
- Disable TARP Caching
- Disable TARP PDU Origination and Propagation
- Configure Multiple NSAP Addresses
- Configure Static TARP Adjacency and Blacklist Adjacency
- Determine TIDs and NSAPs
- Configure TARP Timers
- Configure Miscellaneous TARP PDU Information
- Monitor and Maintain the TARP Protocol

Enable TARP and Configure a TARP TID

TARP must be explicitly enabled before the TARP functionality be comes available and the router must have a TID assigned. Also, before TARP packets can be sent out on an interface, each interface must have TARP enabled and the interface must be able to propagate TARP PDUs.

The router will use the CLNS capability to send and receive TARP PDUs. If the router is configured as an IS, the router must be running IS-IS. If the router is configured as an ES, the router must be running ES-IS.

1. In global configuration mode, turn on the TARP functionality:

- Turn on the TARP functionality.

```
tarp run
```

- Assign a TID to the router.

```
tarp tid tid
```

2. In interface configuration mode, enable TARP on one or more interfaces:

```
tarp enable
```

Disable TARP Caching

By default, TID-to-NSAP address mappings are stored in the TID cache. Disabling this capability clears the TID cache. Re-enabling this capability restores any previously cleared local entry and all static entries.

In global configuration mode, disable TID-to-NSAP address mapping in the TID cache:

```
no tarp allow-caching
```

Disable TARP PDU Origination and Propagation

By default, the router originates TARP PDUs and propagates TARP PDUs to its neighbors, and the interface propagates TARP PDUs to its neighbor. Disabling these capabilities means that the router no longer originates TARP PDUs. Also, the router and the specific interface no longer propagate TARP PDUs received from other routers.

- In global configuration mode, disable origination of TARP PDUs:

  ```
  no tarp originate
  ```

- In global configuration mode, disable global propagation of TARP PDUs.

  ```
  no tarp global-propagate
  ```

- In interface configuration mode, disable propagation of TARP PDUs on a specific interface:

  ```
  no tarp propagate
  ```

Configure Multiple NSAP Addresses

A router may have more than one NSAP address. When a request for an NSAP is sent (Type 1 or Type 2 PDU), the first NSAP address is returned. To receive all NSAP addresses associated with the router, enter a TID-to-NSAP static route in the TID cache for each NSAP address.

In global configuration mode, create a TID-to-NSAP static route:

```
tarp map tid nsap
```

Configure Static TARP Adjacency and Blacklist Adjacency

In addition to all its IS-IS/ES-IS adjacencies, a TARP router propagates PDUs to all its static TARP adjacencies. If a router is not running TARP, the router discards TARP PDUs rather than propagating the PDUs to all its adjacencies. To allow TARP to bypass routers enroute that may not have TARP running, TARP provides a static TARP adjacency capability. Static adjacencies are maintained in a special queue.

In global configuration mode, create a static TARP adjacency:

```
tarp route-static nsap
```

To stop TARP from propagating PDUs to an IS-IS/ES-IS adjacency that may not have TARP running, TARP provides a blacklist adjacency capability. The router will not propagate TARP PDUs to blacklisted routers.

In global configuration mode, bypass a router not running TARP.

```
tarp blacklist-adjacency nsap
```

Determine TIDs and NSAPs

In EXEC mode, determine an NSAP address for a TID or a TID for an NSAP address:

1. Get the TID associated with a specific NSAP.

   ```
   tarp query nsap
   ```

2. Get the NSAP associated with a specific TID.

   ```
   tarp resolve tid [1 | 2]
   ```

To determine the TID, the router first checks the local TID cache. I there is a TID entry in the local TID cache, the requested information is displayed. If there is no TID entry in the local TID cache, a TARP Type 5 PDU is sent out to the specified NSAP address.

To determine the NSAP address, the router first checks the local TID cache. If there is an NSAP entry in the local TID cache, the requested information is displayed. If there is no NSAP entry in the local TID cache, a TARP Type 1 or Type 2 PDU is sent out. By default, a Typ 1 PDU is sent to all Lev el1 (IS-IS and ES-IS) neighbors. If a response is received, the requested information is displayed. If a response is not received within the response time, a Type 2 PDU is sent to all L evel1 and Leve l2 neighbors. Specifying the EXEC command **`tarp resolve`** ***`tid`*** **`2`** causes only a Type 2 PDU to be sent.

You can configure the length of time that the router will wait for a response (in the form of a Type 3 PDU).

Configure TARP Timers

TARP timers provide default values and typically do not need to be changed.

The amount of time that the router waits to receive a response from a Type 1 PDU, a Type 2 PDU, and a Type 5 PDU can be configured. In addition, you can also configure the PDU's lifetime based on the number of hops.

Timers can also be set to control how long dynamically created TARP entries remain in the TID cache, and how long the system ID-to-sequence number mapping entry remains in the loop detection buffer table. The loop detection buffer table prevents TARP PDUs from looping.

In global configuration mode, configure TARP PDU timers, control PDU lifetime, and set how long entries remain in cache:

1. Configure the number of seconds that the router will wait for a response from a TARP Type 1 PDU.

   `tarp t1-response-timer` ***`seconds`***

2. Configure the number of seconds that the router will wait for a response from a TARP Type 2 PDU.

   `tarp t2-response-timer` ***`seconds`***

3. Configure the number of seconds that the router will wait for a response from a TARP Type 2 PDU after the default timer has ex pired.

   `tarp post-t2-response-timer` ***`seconds`***

4. Configure the number of seconds that the router will wait for a response from a TARP Type 5 PDU.

   `tarp arp-request-timer` ***`seconds`***

5. Configure the number of routers that a TARP PDU can traverse before it is discarded.

   `tarp lifetime` ***`hops`***

6. Configure the number of seconds a dynamically-created TARP entry remains in the TID cache.

   `tarp cache-timer` ***`seconds`***

7. Configure the number of seconds that a system ID-to-sequence number mapping entry remains in the loop detection buffer table.

   `tarp ldb-timer` ***`seconds`***

Configure Miscellaneous TARP PDU Information

TARP default PDU values typically do not need to be changed.

You can configure the sequence number of the TARP PDU, set the update remote cache bit used to control whether the remote router up dates its cache, specify the N-selector used in the PDU to indicate TARP PDU, and specify the network protocol type used in outgoing PDUs.

In global configuration mode, configure miscellaneous PDU information:

1. Change the sequence number in the next outgoing TARP PDU.

   `tarp sequence-number` ***`number`***

2. Set the update remote cache bit in all subsequent outgoing TARP PDUs so that the remote router does or does not update the cache.

   `tarp urc [0 | 1]`

3. Specify the N-selector used to identify TARP PDUs.

   `tarp` ***`selector hex-digit`***

4. Specify the protocol type used in outgoing TARP PDUs. Only FE (to indicate CLNP) is supported.

   `tarp` ***`protocol hex-digit`***

Monitor and Maintain the TARP Protocol

Use these EXEC commands to monitor and maintain the TARP caches, tables, and databases:

- Reset the TARP counters that are shown with th **`show tarp traffic`** command.

  `clear tarp counters`

- Remove all system ID-to-sequence number mapping entries in the TARP loop detection buffer table.

  `clear tarp ldb-table`

- Remove all dynamically created TARP TID-to-NSAP address mapping entries in the TID cache.

  `clear tarp tid-table`

- Display all global TARP parameters.

  `show tarp`

- List all adjacencies that are blacklisted (that is, adjacencies that will not receive propagated TARP PDUs).

  `show tarp blacklisted-adjacencies`

- Display information about a specific TARP router stored in the local TID cache.

  `show tarp host` ***`tid`***

- List all interfaces on the router that have TARP enabled.

  `show tarp interface [` ***`type number`*** **`]`**

- Display the contents of the loop detection buffer table.

  `show tarp ldb`

- List all the static entries in the TID cache.

  `show tarp map`

- List all static TARP adjacencies.

  `show tarp static-adjacencies`

- Display information about the entries in the TID cache.

  `show tarp tid-cache`

- Display statistics about TARP PDUs.

  `show tarp traffic`

Examples: TARP Configuration

This section provides both basic and complex examples of TARP configuration.

Basic TARP Configuration Example

This example enables TARP on the router and interface Ethernet 0. The router is assigned the TID myname.

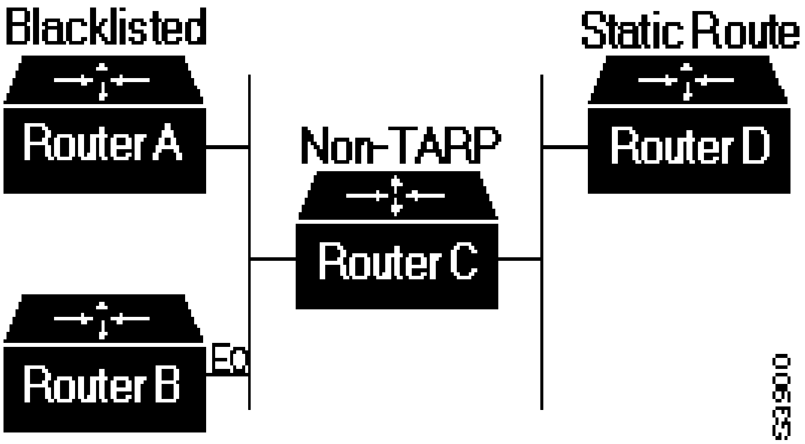
```
clns routing
tarp run
tarp tid myname

interface ethernet 0
tarp enable
```

Complex TARP Configuration Example

Figure 9-13 and this example show how to enable TARP on Router A and on interface Ethernet 0 and assign the TID myname. A static rout is created from Router A (49.0001.1111.1111.00) to Router D (49.0004.1234.1234.1234.00) so that Router D can receive TARP PDUs because Router C in not TARP capable. A blacklist adjacency is also created on Router A for Router B (49.0001.7777.7777.7777.00 so that Router A does not send any TARP PDUs to Router B.

Figure 9-13:Sample TARP Configuration



```
clns routing
tarp run
tarp cache-timer 300
tarp route-static 49.0004.1234.1234.1234.00
tarp blacklist-adjacency 49.0001.7777.7777.7777.00
tarp tid myname

interface ethernet 0
tarp enable
!=
```

Chapter 10: Serial Interface Configuration Steps

Introduction

This chapter describes how to configure synchronous serial interfaces.

Configure the Synchronous Serial Interfaces

The synchronous serial interfaces are configured to allow connection to WANs. Once the Ethernet or Token Ting port on your router has been configured, complete these steps to configure the synchronous serial interfaces:

1. Press <Enter> or type yes to configure serial port 0:

```
Configuring interface Serial0:
  Is this interface in use? [yes]:
```

2. Determine which protocols you want on the synchronous serial interface and enter the appropriate responses:

```
Configure IP on this interface? [yes]:
Configure IP unnumbered on this interface? [no]:
   IP address for this interface: 172.16.73.1
   Number of bits in subnet field [8]:
   Class B network is 172.16.0.0, 8 subnet bits; mask is

   255.255.255.0
Configure AppleTalk on this interface? [no]: yes
   Extended AppleTalk network? [yes]:
   AppleTalk starting cable range [2]: 4

   AppleTalk ending cable range [3]: 4
   AppleTalk zone name [myzone]: ZZ Serial
   AppleTalk additional zone name:
Configure IPX on this interface? [no]: yes
   IPX network number [2]: B002
```

3. Configure the second synchronous serial interface, for example:

```
Configuring interface Serial 1:
   Is this interface in use? [yes]:
   Configure IP on this interface? [yes]:
   Configure IP unnumbered on this interface? [no]:
     IP address for this interface: 172.16.74.2
     Number of bits in subnet field [8]:
     Class B network is 172.16.0.0, 8 subnet bits; mask is

     255.255.255.0
   Configure AppleTalk on this interface? [no]: yes
     AppleTalk starting cable range [3]: 5
     AppleTalk ending cable range [4]: 5
     AppleTalk zone name [myzone]: ZZ Serial
     AppleTalk additional zone name:
   Configure IPX on this interface? [no]: yes
     IPX network number [3]: B003
```

4. The configuration you entered now displays and you are asked if you want to use the displayed configuration. If you enter no, you will lose the configuration information you just entered and you can begin the configuration again. If you type yes, the configuration will be entered and saved in the startup configuration:

```
Use this configuration? [yes/no]: yes
Building configuration...
Use the enabled mode 'configure' command to modify this configuration.
Press RETURN to get started!
```

Chapter 11: AI2524 Sync PPP Configuration Steps

Introduction

This chapter describes how to enable PPP encapsulation and perform a variety of PPP configuration tasks.

Configuration Overview

The Point-to-Point Protocol (PPP), described in RFCs 1661 and 1332, encapsulates network layer protocol information over point-to-point links. You can configure PPP on these physical interfaces:

- Asynchronous serial
- HSSI
- ISDN
- Synchronous serial

The software provides PPP as an encapsulation method. It also provides the Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) on serial interfaces running PPP encapsulation. By enabling PPP encapsulation on physical inter faces, PPP can also be used on calls placed by dialer interfaces that use physical interfaces.

The current implementation of PPP supports option 3, authentication using CHAP or PAP, option 4, Link Quality Monitoring, and option 5, Magic Number configuration options. The software always sends option 5 and negotiates for options 3 and 4 if so configured. All other options are rejected.

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how th **`down-when-looped`** command is configured, the router might shut down a link if it detects a loop.

The AI2524 supports the following upper-layer protocols: Bridging, CLNS, IP, and IPX.

PPP Configuration Task List

To configure PPP on a serial interface, you must enable PPP encapsulation.

You can also complete these tasks; these tasks are optional but offer variety of uses and enhancements for PPP on your systems and networks:

- Enable CHAP or PAP Authentication
- Enable Link Quality Monitoring (LQM)
- Configure Automatic Detection of Encapsulation Type
- Configure Compression of PPP Dat
- Configure IP Address Pooling
- Configure PPP Callback
- Disable or Reenable Peer Neighbor Routes
- Configure PPP Half-Bridging
- Configure Multilink PPP
- Configure Virtual Private Dial-up Networks
- Enable PPP on VTY Lines for Asynchronous Access over ISDN
- Monitor and Maintain MLP, MMP, and VPDN Virtual Interfaces

Enable PPP Encapsulation

You can enable PPP on serial lines to encapsulate IP and other network protocol datagrams in interface configuration mode:

```
encapsulation ppp
```

PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. The**no keepalive** command can be used to disable echo requests.

Enable CHAP or PAP Authentication

The Point-to-Point Protocol (PPP) with Challenge Handshake Authentication Protocol (CHAP) authentication or Password Authentication Protocol (PAP) is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP are specified in RFC 1334. These protocols are sup ported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a name. This identification process prevents a route from placing another call to a router to which it is already connected and prevents unauthorized access.

Access control using CHAP or PAP is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note: ***To use CHAP or PAP, you must be running PPP encapsulation.***

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or challenges the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

The required response consists of two parts:

- An encrypted version of the ID, a secret password (or secret), and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret by performing the same encryption operation as indicated in the response and looking up the required host name or username. The secret passwords must be identical on the remote device and the local router.

By transmitting this response, the secret is never transmitted in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only at the time a link is established. The local router or access server does not request a password during the rest of the call. The local device can, however, respond to such requests from other devices during a call.

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

1. In interface configuration mode, enable PPP encapsulation:

   ```
   encapsulation ppp
   ```

2. In interface configuration mode, enable CHAP or PAP authentication on an interface configured for PPP encapsulation:

   ```
   ppp authentication {chap | chap pap | pap
   chap | pap} [if-needed] [ list-name |
   default] [callin]
   ```

   The **ppp authentication chap** optional keyword, is used only with TACACS or extended TACACS. The optional keyword **list-name** is used only with AAA/TACACS+.

Note: ***If you use a*** ***list-name*** ***that has not been configured with the*** **aaa authentication ppp** ***command, you disable PPP on the line.***

3. Add a username entry for each remote system from which the local router or access server requires authentication.

   In global configuration mode, specify the password to be used in CHAP or PAP caller identification:

   ```
   username name password secret
   ```

4. In interface configuration mode, configure Terminal Access Controller Access Control System (TACACS) on a specific interface as an alternative to global host authentication:

   ppp use-tacacs [single-line]

   or

   aaa authentication ppp

   Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with Authentication, Authorization, and Accounting (AAA)/TACACS+.

Example: CHAP with an Encrypted Password

These configuration examples enable CHAP on interface serial 0 of three devices.

1. Configure Router yyy.

```
hostname yyy
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

2. Configure Router xxx.

```
hostname xxx
interface serial 0
encapsulation ppp
ppp authentication chap
username yyy password secretxy
username zzz password secretxz
```

3. Configure Router zzz.

```
hostname zzz
interface serial 0
encapsulation ppp
ppp authentication chap
username xxx password secretxz
username yyy password secretzy
```

When you look at the configuration file, the passwords will be en crypted and the display will look similar to:

```
hostname xxx
interface serial 0
encapsulation ppp
ppp authentication chap
username yyy password 7 121F0A
username zzz password 7 1329A05
```

Enable Link Quality Monitoring (LQM)

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the router shuts down the link. The percentages are calculated for both the incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent to the total number of packets and bytes received by the destination node. The incoming quality is calculated by comparing the total number of packets and bytes received to the total number of packets and bytes sent by the destination peer.

When LQM is enabled, Link Quality Reports (LQRs) are sent every keepalive period. LQRs are sent in place of keepalives. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

In interface configuration mode, enable LQM on the interface:

ppp quality ***percentage***

The percentage argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and taken down.

Configure Automatic Detection of Encapsulation Type

You can enable a serial or ISDN interface to accept calls and dynamically change the encapsulation on the interface when the remote device does not signal the call type. For example, if an ISDN call does not identify the call type in the Lower Layer Compatibility fields and is using an encapsulation that is different from the one configured on the interface, the interface can change its encapsulation type on the fly.

This feature enables interoperation with ISDN terminal adapters that use V.120 encapsulation but do not signal V.120 in the call setup message. An ISDN interface that, by default, answers a call as synchronous serial with PPP encapsulation can change its encapsulation and answer such calls.

Automatic detection is attempted for the first 10 seconds after the link is established or for the first 5 packets exchanged over the link, whichever is first.

In interface configuration mode, enable automatic detection of encapsulation type:

```
autodetect encapsulation encapsulation-type
```

You can specify one or more encapsulations to detect. Cisco IOS software currently supports automatic detection of PPP and V.120 encapsulations.

Configure Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. The compression algorithm used is a predictor algorithm (the RAND algorithm), which uses a compression dictionary to predict the next character in the frame.

PPP encapsulations support both predictor and Stacker compression algorithms.

Compression is performed in software and might significantly affect system performance. Disable compression if the router CPU load ex ceeds 65%. To display the CPU load, use th **show process cpu** EXEC command.

If the majority of your traffic is already compressed files, do not us compression.

In interface configuration mode, configure compression over PPP:

1. Enable encapsulation of a single protocol on the serial line.

   ```
   encapsulation ppp
   ```

2. Enable compression.

```
ppp compress [predictor | stac]
```

Configure IP Address Pooling

Point-to-point interfaces must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be:

- Configured through the command lin
- Entered with an EXEC-level command
- Provided by TACACS+, DHCP, or from a locally administered pool

IP address pooling consists of a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through the IP Control Protocol (IPCP) address negotiation process. It also enhances the flexibility of configuration by allowing multiple types of pooling to be active simultaneously.

IP address pooling allows the configuration of a global default address pooling mechanism, per-interface configuration of the mechanism, and per-interface configuration of a specific address or pool name.

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup. This method is used only if the peer requests an IP address, if no other peer IP address has been assigned, and if the interface is a member of a dialer group.
- PPP or SLIP EXEC command. An asynchronous dial-up user can enter a peer IP address or host name when PPP or SLIP is invoked from the command line. The address is used for the current session and then discarded.
- IPCP negotiation. If the peer presents a peer IP address during IPCP address negotiation and if no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Chat script. The IP address in the **`dialer map`** command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- VTY/Protocol translation. Th **`translate`** command can defin the peer IP address for a VTY (pseudo async interface).

- Default IP address. The **`peer default ip address`** command and the **`member peer default ip address`** command can be used to define default peer IP addresses.
- TACACS+ assigned IP address. During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that can be used by the user being authenticated on a dial-up interface. This address overrides any default IP address and prevents pooling from taking place.
- DHCP retrieved IP address. If configured, the routers acts as a proxy client for the dial-up user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.
- Local address pool. The local address pool contains a set of contiguous IP addresses (a maximum of 256 addresses) stored in two queues. The free queue contains addresses available to be assigned. The used queue contains addresses that are in use. Addresses are stored in the free queue in First-In First-Out (FIFO order to minimize the chance the address will be reused and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.

  The pool configured for the interface is used, unless TACACS+ returns a pool name as part of authentication, authorization, and accounting (AAA). If no pool is associated with a given interface, the global pool named default is used.

Precedence Rules

These precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address ex ists)

4. Address from an EXEC-level PPP or SLIP command or from a chat script

5. Configured address from the **`peer default ip address`** command or address from the protocol **`translate`** command

6. Peer provided address from IPCP negotiation (not accepted unless no other address exists)

Interfaces Affected

IP address pooling is available on all asynchronous serial, synchro nous serial, ISDN BRI, and ISDN PRI interfaces running the Point-to-Point Protocol (PPP) or Serial Line Internet Protocol (SLIP).

Choose the IP Address Assignment Method

IP address pooling allows configuration of a global default address pooling mechanism, per-interface configuration of the mechanism, and per-interface configuration of a specific address or pool name.

You can define the type of IP address pooling mechanism used on router interfaces by completing these actions:

- Define the Global Default Mechanism
- Configure Per-Interface IP Address Assignment

Define the Global Default Mechanism

The Global Default Mechanism applies to all point-to-point interfaces (asynchronous, synchronous, ISDN BRI, ISDN PRI, and dialer interfaces) that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the Global Default Mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform one of these tasks:

- Define DHCP as the Global Default Mechanism
- Define Local Address Pooling as the Global Default Mechanism

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface, or you can configure the interface with a specifi IP address to be used for dial-in peers.

Define DHCP as the Global Default Mechanism

The Dynamic Host Configuration Protocol (DHCP) specifies these components:

- A DHCP server. A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy-client. An access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. Th DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

In global configuration mode, enable DHCP as the global default mechanism:

1. Specify DHCP client-proxy as the global default mechanism.

   `ip address-pool dhcp-proxy-client`

2. (Optional) Specify the IP address of a DHCP server for the proxy client to use.

   ip dhcp-server [*ip-address* | *name*]

You can provide up to ten DHCP servers for the proxy-client (the router or access server) to use. DHCP servers provide temporary IP addresses.

Define Local Address Pooling as the Global Default Mechanism

In global configuration mode, define local pooling as the global default mechanism:

1. Specify local pooling as the global default mechanism.

   `ip address-pool local`

2. Create one or more local IP address pools.

   ip local pool {default | *poolname*} *low-ip-address* [*high-ip-address*]

If no other pool is defined, the local pool called default is used.

Configure Per-Interface IP Address Assignment

When you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can then configure the few interfaces for which it is important to have a nondefault configuration. You can do any of these:

1. In global configuration mode, define a nondefault address pool for use on an interface:

- Create one or more local IP address pools.

  ip local pool ***poolname*** **{*low-ip-address* [*high-ip-address*]}**

- Specify the interface and enter interface configuration mode.

  interface ***type number***

- Specify the pool for the interface to use.

  peer default ip address pool ***poolname***

2. In global configuration mode, define DHCP as the IP address mechanism for an interface:

- Specify the interface and enter interface configuration mode.

  interface ***type number***

- Specify DHCP as the IP address mechanism on this interface.

  peer default ip address pool dhcp

3. Beginning in global configuration mode, define a specific IP ad dress to be assigned to all dial-in peers on an interface:

- Specify the interface and enter interface configuration mode.

  interface ***type number***

- Specify the IP address to assign.

  peer default ip address ***ip-address***

4. Beginning in global configuration mode, make temporary IP addresses available on a per-interface basis for dial-in asynchronous clients using Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP):

- Specify that the access server use a local IP address pool on all asynchronous interfaces.

  ip address-pool local

- Create one or more local IP address pools.

  ip local pool {default | ***poolname*** **{** ***begin-ip-address-range*** **[*****end-ip-address-range*****]}}**

- (Optional) Enter interface configuration mode.

  interface async ***number***

- (Optional) If you want an interface to use an address pool other than default, specify which pool each interface uses.

  peer default ip address pool ***poolname***

Configure PPP Callback

PPP callback provides a client-server relationship between the end points of a point-to-point connection. PPP callback allows a router to request that a dial-up peer router call back. The callback feature can be used to control access and toll costs between the routers.

When PPP callback is configured on the participating routers, the calling router (the callback client) passes authentication information to th remote router (the callback server), which uses the host name and dial string authentication information to determine whether to place a return call. If the authentication is successful, the callback server disconnects and then places a return call. The remote username of the return call is used to associate it with the initial call so that packets can b transmitted.

Both routers on a point-to-point link must be configured for PPP callback; one must function as a callback client and one must be configured as a callback server. The callback client must be configured to initiate PPP callback, and the callback server must be configured to accept PPP callback.

This feature implements these callback specifications of RFC 1570:

- For the client. Option 0, location is determined by user authentication
- For the server. Option 0, location is determined by user authentication; Option 1, dialing string; and Option 3, E.164 number.

Return calls are made through the same dialer rotary group but not necessarily the same line as the initial call.

Note: ***If the return call fails (because the line is not answered or the line is busy), no retry occurs. If the callback server has no interface available when attempting the return call, it does not retry.***

Configure a Router as a Callback Client

Beginning in global configuration mode, configure a router interface as a callback client:

1. Specify the interface.

   ```
   interface serial number
   ```

2. Enable DDR. Set parity on synchronous serial interfaces and asynchronous interfaces.

   ```
   dialer in-band [no-parity | odd-parity]
   ```

3. Enable PPP encapsulation.

   ```
   encapsulation ppp
   ```

4. Enable CHAP or PAP authentication.

   ```
   ppp authentication {chap | pap}
   ```

5. Map the next hop address to the host name and phone number.

   ```
   dialer map protocol next-hop-address name
   hostname dial-string
   ```

6. Enable the interface to request PPP callback for this callback map class.

   ```
   ppp callback request
   ```

7. (Optional) Configure a dialer hold queue to store packets for this callback map class.

   ```
   dialer hold-queue packets timeout seconds
   ```

Example: PPP Callback Client

The PPP callback client is configured on an ISDN BRI interface in a router in Dallas. The callback client does not require an enable timeout and a map class to be defined.

```
interface BRI0
 ip address 7.1.1.8 255.255.255.0
 encapsulation ppp
 dialer map ip 7.1.1.7 name dallas 81012345678902
 dialer-group 1
 ppp callback request
 ppp authentication chap
```

Configure a Router as a Callback Server

Beginning in global configuration mode, configure a router as a call back server:

1. Specify the interface and enter interface configuration mode.

   interface serial ***number***

2. Enable DDR. Set parity on synchronous serial interfaces and asynchronous interfaces.

   dialer in-band [no-parity | odd-parity]

3. Enable PPP encapsulation.

   encapsulation ppp

4. Enable CHAP or PAP authentication.

   ppp authentication {chap | pap}

5. Map the next hop address to the host name and phone number, using the name of the map class established for PPP callback on this interface.

   dialer map ***protocol address*** **name** ***hostname*** **class** ***classname dial-string***

6. (Optional) Configure a dialer hold queue to store packets to be transferred when the callback connection is established.

   dialer hold-queue ***number*** **timeout** ***seconds***

7. (Optional) Configure a timeout period between calls.

   dialer enable-timeout ***seconds***

8. Configure the interface to accept PPP callback.

   ppp callback accept

9. (Optional) Enable callback security, if desired.

   dialer callback-secure

10. Return to global configuration mode.

    exit

11. Configure a dialer map class for PPP callback.

    map-class dialer ***classname***

12. Configure a dialer map class as a callback server.

    `dialer callback-server [`*`username`*`]`

The enable timer default is 15 seconds. The time between the initial call and the return call can be improved by reducing this number, but care should be taken to ensure that the initial call is completely disconnected before the timer expires.

Note: ***On the PPP callback server, the dialer enable-timeout functions as the timer for returning calls to the callback client.***

Example: PPP Callback Server

The PPP callback server is configured on an ISDN BRI interface in router in Atlanta. The callback server requires an enable timeout and map class to be defined.

```
interface BRI0
 ip address 7.1.1.7 255.255.255.0
 encapsulation ppp
 dialer callback-secure
 dialer enable-timeout 2
 dialer map ip 7.1.1.8 name atlanta class dial1 81012345678901
 dialer-group 1
 ppp callback accept
 ppp authentication chap
!
map-class dialer dial1
dialer callback-server username
```

Disable or Reenable Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on point-to-point interface when the PPP IPCP negotiation is completed.

In interface configuration mode, disable this default behavior o reenable it once it has been disabled:

- Disable creation of neighbor routes.

  ```
  no peer neighbor-route
  ```

- Reenable creation of neighbor routes.

  ```
  peer neighbor-route
  ```

Note: ***If entered on a dialer or async-group interface, this command affects all member interfaces.***

Configure PPP Half-Bridging

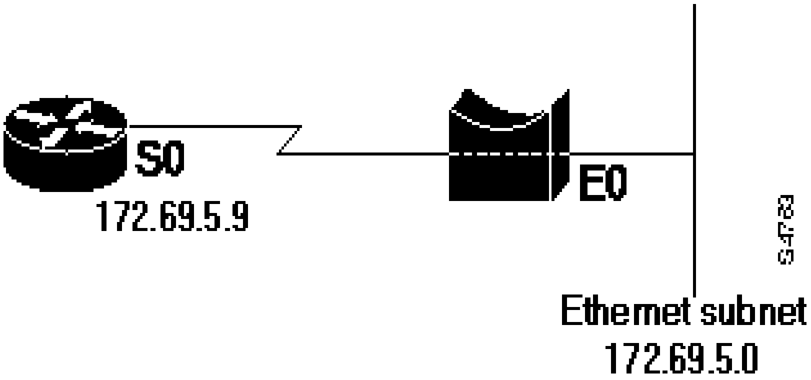
For situations in which a routed network needs connectivity to a remote bridged Ethernet network, a serial or ISDN interface can be configured to function as a PPP half-bridge. The line to the remote bridge functions as a virtual Ethernet interface. The router's serial or ISDN interface functions as a node on the same Ethernet subnetwork as the remote network.

The bridge sends bridge packets to the PPP half-bridge, which converts them to routed packets and forwards them to other router pro cesses. Likewise, the PPP half-bridge converts routed packets to Ethernet bridge packets and sends them to the bridge on the sam Ethernet subnetwork.

Note: ***An interface cannot function as both a half-bridge and a bridge.***

Figure 11-1 shows a router with a serial interface configured as a PPP half-bridge. The interface functions as a node on the Ethernet subnetwork with the bridge. Note that the serial interface has an IP address on the same Ethernet subnetwork as the bridge.

Figure11-1: Router Serial Interface Configured as a Half-Bridge



Note: ***The Cisco IOS software supports no more than one PPP half-bridge per Ethernet subnetwork.***

Beginning in global configuration mode, configure a serial interface to function as a half-bridge:

1. Specify the interface and enter interface configuration mode.

   `interface serial` *`number`*

2. Enable PPP half-bridging for one or more routed protocols:

   `ppp bridge ip`

   `ppp bridge ipx [novell-ether | arpa | sap | snap]`

3. Provide a protocol address on the same subnetwork as the remote network.

   `ip address` *`n.n.n.n`*

   `ipx network` *`network`*

Note: ***You must enter the `ppp bridge` command either when the interface is shut down or before you provide a protocol address for the interface.***

Configure Multilink PPP

Multilink PPP (MLP) provides load balancing over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic.

Multilink PPP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the sam remote address. The multiple links come up in response to a dialer load threshold that you define. The load can be calculated on inbound traffic, outbound traffic, or on either. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

Multilink PPP is designed to work over single or multiple interfaces that are configured to support both dial-on-demand rotary groups and PPP encapsulation:

- Asynchronous serial interfaces
- Basic Rate Interfaces (BRIs)
- Primary Rate Interfaces (PRIs)

Configure Multilink PPP on Asynchronous Interfaces

Beginning in global configuration mode, configure an asynchronous interface to support DDR and PPP encapsulation:

1. Specify an asynchronous interface.

   `interface async` ***`number`***

2. Specify no IP address for the interface.

   `no ip address`

3. Enable PPP encapsulation.

   `encapsulation ppp`

4. Enable DDR on the interface.

   `dialer in-band`

5. Include the interface in a specific dialer rotary group.

   `dialer rotary-group` ***`number`***

Repeat these steps for additional asynchronous interfaces, as needed.

At some point, adding more asynchronous interfaces does not improve performance. With the default maximum transmission unit (MTU) size, Multilink PPP should support three asynchronous interfaces using V.34 modems. However, packets might be dropped occasionally if the MTU is small or if large bursts of short frames occur.

Beginning in global configuration mode, configure a dialer interface to support PPP encapsulation and Multilink PPP:

1. Define a dialer rotary group.

   interface dialer ***number***

2. Specify no IP address for the interface.

   no ip address

3. Enable PPP encapsulation.

   encapsulation ppp

4. Enable DDR on the interface.

   dialer in-band

5. Configure bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.

   dialer load-threshold ***load*** **[inbound | outbound | either]**

6. Enable Multilink PPP.

   ppp multilink

Configure Multilink PPP on a Single ISDN BRI Interface

To enable Multilink PPP on a single Integrated Services Digital Net work (ISDN) BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

Beginning in global configuration mode, enable PPP on an ISDN BRI interface:

1. Specify an interface.

   interface bri ***number***

2. Provide an appropriate protocol address for the interface.

   ip address ***ip-address mask***

3. Enable PPP encapsulation.

   encapsulation ppp

4. (Optional) Specify a dialer idle timeout.

   dialer idle-timeout ***seconds***

5. Specify the dialer load threshold for bringing up additional WAN links.

   dialer load-threshold ***load***

6. Configure the ISDN interface to call the remote site.

   dialer map ***protocol next-hop-address*** **[name** ***hostname*****] [spc] [speed 56 | 64] [broadcast] [*****dial-string*****[:*****isdn-subaddress*****]]**

7. Add the interface to a dialer rotary group.

   dialer-group ***group-number***

8. (Optional) Enable PPP authentication.

   ppp authentication pap

9. Enable Multilink PPP on the dialer rotary group.

   ppp multilink

If you do not use PPP authentication procedures, your telephone service must pass caller ID information. The load threshold number is required.

When Multilink PPP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer.

The **dialer-load threshold 1** command does not keep a multilink bundle of n links connected indefinitely, and the **dialer-load threshold 2** command does not keep a multilink bundle of two links connected indefinitely.

Example: Multilink PPP on One ISDN Interface

This example enables Multilink PPP on the BRI 0 interface. Because an ISDN interface is a rotary group by default, when one BRI is configured, no dialer rotary group configuration is required.

```
interface bri 0
description connected to ntt 81012345678902
ip address 7.1.1.7 255.255.255.0
encapsulation ppp
dialer idle-timeout 30
dialer load-threshold 40 either
dialer map ip 7.1.1.8 name atlanta 81012345678901
dialer-group 1
ppp authentication pap
ppp multilink
```

Configure Multilink PPP on Multiple ISDN BRI Interfaces

To enable Multilink PPP on multiple ISDN BRI interfaces, set up a dialer rotary interface and configure it for Multilink PPP and then con figure the BRIs separately and add them each to the same rotary group.

Beginning in global configuration mode, set up the dialer rotary interface for the BRI interfaces:

1. Specify the dialer rotary interface.

   `interface dialer` ***`number`***

2. Specify the protocol address for the dialer rotary interface.

   `ip address` ***`address mask`***

3. Enable PPP encapsulation.

   `encapsulation ppp`

4. Specify in-band dialing.

   `dialer in-band`

5. Specify the dialer idle timeout period, using the same timeout period as the individual BRI interfaces.

   `dialer idle-timeout` ***`seconds`***

6. Map the next-hop protocol address and name to the dial string needed to reach it.

   dialer map *protocol next-hop-address* [name *hostname*] [spc] [speed 56 | 64] [broadcast] [*dial-string*[:*isdn-subaddress*]]

7. Specify the dialer load threshold, using the same threshold as the individual BRI interfaces.

   dialer load-threshold *load*

8. Control access to this interface by adding it to a dialer access group.

   dialer-group *group-number*

9. (Optional) Enable PPP CHAP authentication.

   ppp authentication chap

10. Enable Multilink PPP.

    ppp multilink

If you do not use PPP authentication procedures, your telephone service must pass caller ID information.

Beginning in global configuration mode, configure each of the BRIs to belong to the same rotary group:

1. Specify one of the BRI interfaces.

   interface bri *number*

2. Specify that it does not have an individual protocol address.

   no ip address

3. Enable PPP encapsulation.

   encapsulation ppp

4. Set the dialer idle timeout period, using the same timeout for each of the BRI interfaces you configure.

   dialer idle-timeout *seconds*

5. Add the interface to the rotary group.

   dialer rotary-group *group-number*

6. Specify the dialer load threshold for bringing up additional WAN links.

   ```
   dialer load-threshold  load
   ```

Repeat Steps 1 through 6 for each BRI you want to belong to the same dialer rotary group.

When Multilink PPP is configured and you want a multilink bundle to be connected indefinitely, use the **`dialer idle-timeout`** command to set a very high idle timer.

The **`dialer load-threshold 1`** command does not keep a multilink bundle of n links connected indefinitely, and the **`dialer load-threshold 2`** command does not keep a multilink bundle of two links connected indefinitely.

Example: Multilink PPP on Multiple ISDN Interfaces

This example configures multiple ISDN BRIs to belong to the same dialer rotary group for Multilink PPP. Th **`dialer rotary-group`** command is used to assign each of the ISDN BRIs to that dialer rotary group.

```
interface BRI0
no ip address
encapsulation ppp
dialer idle-timeout 500
dialer rotary-group 0
dialer load-threshold 255 balanced
!
interface BRI1
no ip address
encapsulation ppp
dialer idle-timeout 500
dialer rotary-group 0
dialer load-threshold 255 balanced
!
interface BRI2
no ip address
encapsulation ppp
dialer idle-timeout 500
dialer rotary-group 0
dialer load-threshold 255 balanced
!
interface Dialer0
ip address 99.0.0.2 255.0.0.0
encapsulation ppp
dialer in-band
dialer idle-timeout 500
dialer map ip 99.0.0.1 name atlanta broadcast 81012345678901
dialer load-threshold 255 balanced
dialer-group 1
ppp authentication chap
ppp multilink
```

Configure Virtual Private Dial-up Networks

Virtual private dial-up networks (VDPN) allow separate and autonomous protocol domains to share common access infrastructure including modems, access servers, and ISDN routers. VPDN uses the Level 2 Forwarding protocol (L2F) which permits the tunneling of link level frames.

Using L2F tunneling, an Internet Service Provider (ISP) or other ac cess service can create a virtual tunnel to link a customer's remote sites or remote users with corporate home networks. In particular, a network access server at the ISP's Point of Presence (POP) exchanges PPP messages with the remote users and communicates by L2F requests and responses with the customer's home gateway to set up tunnels.

L2F passes protocol-level packets through the virtual tunnel between endpoints of a point-to-point connection.

Frames from the remote users are accepted by the ISP POP, stripped of any linked framing or transparency bytes, encapsulated in L2F, and forwarded over the appropriate tunnel. The customer's home gateway accepts these L2F frames, strips the L2F encapsulation, and processes the incoming frames for the appropriate interface.

Note: ***This implementation of VPDN supports PPP dial-up only.***

To configure virtual private dial-up networks, complete these tasks:

- Understand VPDNs
- Beginning in global configuration mode, configure a virtual template for interfaces on a home gateway access server:
- Configure Incoming VPDN Connections on the Home Gateway
- Configure Outgoing VPDN Connections on the Network Access Server

Understand Virtual Private Dial-up Networks

VPDN enables users to configure secure networks that take advantag of internet service providers that tunnel the company's remote access traffic through the ISP cloud.

Remote offices or mobile users can connect to their home network using local dial-up services of third parties. The dial-up service provider agrees to forward the company's traffic from the ISP POP to a company-run home gateway. Network configuration and security remain in the control of the client. The dial-up service provider provides a virtual pipe between the company's sites.

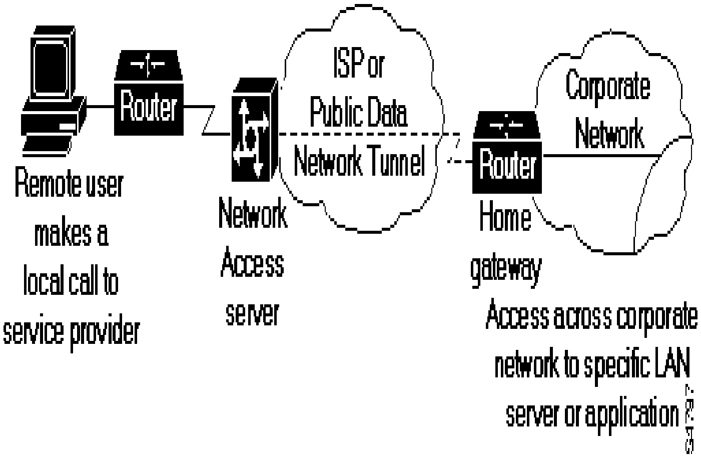
Note: ***The MMP feature uses VPDN to connect multiple PPP sessions for which individual dial-in calls have arrived on different stack group members. VPDN provides speed and reliability for the setup and shutdown of Multilink PPP.***

Complete these steps to create a a VPDN connection between a remote user and the home LAN:

1. The remote user initiates a PPP connection to the ISP using the analog telephone system or ISDN.
2. The ISP network access server accepts the connection.
3. The ISP network access server authenticates the end user with CHAP or PAP. The username is used to determine whether the user is a VPDN client. If the user is not a VPDN client, the client accesses the Internet or other contacted service.
4. The tunnel endpoints—the network access server and the home gateway—authenticate each other before any sessions are at tempted within a tunnel.
5. If no L2F tunnel exists between the network access server and the remote user’s home gateway, a tunnel is created. Once the tunnel exists, an unused slot within the tunnel is allocated.
6. The home gateway accepts or rejects the connection. Initial setup can include authentication information required to allow the home gateway to authenticate the user.
7. The home gateway sets up a virtual interface. Link-level frames can now pass through this virtual interface through the L2F tunnel.

Figure 11-2 illustrates a VPDN connection from a remote user, who makes a local call, to the corporate network, through an end-to-end L2F tunnel (shown by the dotted line).

Figure 11-2: Configure a Virtual Template and Create a Virtual Template Interface on the Home Gateway


Router
Remote user makes a local call to service provider
Network Access server
ISP or Public Data Network Tunnel
Router
Home gateway
Corporate Network
Access across corporate network to specific LAN server or application

Beginning in global configuration mode, configure a virtual templat for interfaces on a home gateway access server:

1. Specify a default local IP address pool.

   ip local pool default ***ip-address***

2. Create a virtual template interface, and enter interface configuration mode.

   interface virtual-template ***number***

3. Identify the virtual template interface type and number on the LAN.

   ip unnumbered ethernet 0

4. Enable PPP encapsulation on the virtual template interface.

   encapsulation ppp

5. Enable PPP authentication on the virtual template interface.

   ppp authentication chap

Configure Incoming VPDN Connections on the Home Gateway

In global configuration mode, configure virtual private dialup networking on a home gateway router or access server:

1. Enable virtual private networking.

   `vpdn enable`

2. Specify the remote host, the local name to use for authenticating, and the virtual template to use.

   `vpdn incoming` ***`remote-name local-name`***
   `virtual-template` ***`number`***

Configure Outgoing VPDN Connections on the Network Access Server

In global configuration mode, configure a network access server to make outgoing L2F connections to a home gateway for VPDN:

1. Enable virtual private networking.

   `vpdn enable`

2. Specify the remote host that is to accept L2F connections.

   `vpdn outgoing` ***`domain-name local-name`*** **`ip`** ***`ip-address`***

Example: Network Access Server Servicing Multiple Domains

This example provides VDPN configurations for a single NAS and two different gateways. The two gateways are presumably located at two entirely separate companies. The NAS decides which company to forward to based on the domain name that is passed by the user.

The commands also illustrate where to configure the commands **`vpdn outgoing`** (on the network access server) and **`vpdn incoming`** (on a home gateway).

NAS1

```
vpdn enable
vpdn outgoing domain1.com nas1 ip 1.1.1.1
vpdn outgoing domain2.com nas2 ip 2.2.2.2
```

Gateway1—Domain1

```
vpdn enable
vpdn incoming nas1 gateway1 virtual-template 1

interface virtual-template 1
  ip unnumbered Ethernet0
  ppp authentication chap
```

Gateway2—Domain2

```
vpdn enable
vpdn incoming nas2 gateway2 virtual-template 1

int virtual-template 1
  ip unnumbered Ethernet0
  ppp authentication chap
```

Example: NAS Servicing Multiple Domains to the Same Gateway

This example provides configurations for one NAS and one Gateway that might have two parallel tunnels between them. Two different domain names are associated with two different virtual interface configurations.

Users dialing in with domain name domain1.com will be forwarded to the home gateway and be given a virtual-access interface based on virtual template 1. Users dialing in with the domain2.com will be forwarded to the same home gateway and be given a virtual-access interface based on virtual template 2.

NAS 1

```
vpdn enable
vpdn outgoing domain1.com nas1 ip 1.1.1.1
vpdn outgoing domain2.com nas2 ip 1.1.1.1
```

Gateway 1

```
vpdn incoming nas1 gateway virtual-template 1
vpdn incoming nas2 gateway virtual-template 2

interface virtual-template 1
  ip unnumbered Ethernet0
  peer default ip address pool domain1-pool
  ppp authentication chap

interface virtual-template 2
  ip unnumbered Ethernet0
  peer default ip address pool domain2-pool
  ppp authentication chap
```

Example: Using TACACS+ for Forwarding from the NAS

This example provides configurations for an NAS and a public domain TACACS+ server. On the NAS it is only necessary to enable AAA and to use the **`vpdn enable`** command.

Users with structured logins (user@domain.com) will have their domain authorized on the TACACS server and will be forwarded if there is a VPDN entry there. If there is no VPDN entry on the TACACS server, the login process will continue as normal.

NAS

```
aaa new-model
vpdn enable
```

TACACS+ Server

```
vpdn outgoing domain.com nas ip 172.21.9.18
```

Enable PPP on VTY Lines for Asynchronous Access over ISDN

You can configure a router to support asynchronous access over ISDN by globally enabling PPP on VTY lines. PPP is typically enabled on synchronous or asynchronous serial interfaces; however, the Cisco IOS software permits you to configure PPP on VTY lines. This configures the VTY line to support asynchronous access over ISDN from an ISDN terminal to a VTY session on the router.

In global configuration mode, enable asynchronous protocol features on all the router's VTY lines:

```
vty-async
```

This task enables PPP on VTY lines on a global basis on the router. To configure PPP on a per-VTY basis, use th **translate** command.

Monitor and Maintain MLP, MMP, and VPDN Virtual Interfaces

To monitor and maintain virtual interfaces, you can perform any of these tasks:

- Display MLP and MMP bundle information.

  ```
  show ppp multilink
  ```

- Display information about the active L2F tunnels and the L2F message identifiers.

  ```
  show vpdn
  ```

- Display the status of the stack group members.

  ```
  show sgbp
  ```

- Display the current seed bid value.

  ```
  show sgbp queries
  ```

Chapter 12: AI2524 X.25 Configuration Steps

Introduction

This chapter describes how to configure connections through X.25 networks and Link Access Procedure Balanced (LAPB) connections. This section also describes how to create X.29 access lists and profile scripts. For users who only want to configure a simple, reliable serial encapsulation method, se Configure LAPB.

X.25 Configuration

X.25 Configuration Task List

To configure X.25, complete the tasks in one or more of these sections, depending upon the X.25 application or task required for your network. The interface, datagram transport, and routing tasks are divided into sections based generally on how common the feature is and how often it is used.

- Configure Interfac
- Configure Additional X.25 Interface Parameters
- Configure an X.25 Datagram Transport
- Configure Additional X.25 Datagram Transport Features
- Configure X.25 Routing
- Configure Additional X.25 Routing Features
- Configure CMNS Routing
- Create X.29 Access Lists
- Create an X.29 Profile Script
- Configure LAPB

All these features can coexist on an X.25 interface.

Default parameters are provided for X.25 operation; however, you can change the settings to meet the needs of your X.25 network or as defined by your X.25 service supplier. The AI2524 also provides additional configuration settings to optimize your X.25 usage.

Note: ***If you connect a router to an X.25 network, use the parameters set by your network administrator for th connection; these parameters will typically be thos described in the Configure Interface and Modify LAPB Protocol Parameters sections. Also, note that the X.25 Level 2 parameters described in the Modify LAPB Protocol Parameters section affect X.25 Level 3 operations.***

Configure Interface

To configure an X.25 interface, perform these tasks:

- Set the X.25 Mode
- Set the Virtual Circuit Ranges
- Set the Packet Numbering Modulo
- Set the X.121 Address
- Set the Default Flow Control Values

These tasks describe the parameters that are essential for correct X.25 behavior. The first task is required. The others might be required or optional, depending on what the router is expected to do and on the X.25 network.

Set the X.25 Mode

A router using X.25 Level 3 encapsulation can:

- Act as a DTE or DCE protocol device (according to the needs of your X.25 service supplier)
- Use the Internet Engineering Task Force (IETF) standard encapsulation, as specified by RFC 1356.

Because the default serial encapsulation is High-level Data Link Control (HDLC), you must explicitly configure an X.25 encapsulation method.

In interface configuration mode, configure the mode of operation and one of these encapsulation types for a specified interface:

```
encapsulation x25 [dte | dce] [ietf]]
```

Typically, a Public Data Network (PDN) will require attachment as DTE. This requirement is distinct from the hardware interface DTE or DCE identity.

The default mode of operation is DTE, and the default encapsulation method is Cisco's pre-IETF method.

Set the Virtual Circuit Ranges

The X.25 protocol maintains multiple connections over one physical link between a DTE and a DCE. These connections are called virtual circuits or Logical Channels (LCs). X.25 can maintain up to 4095 virtual circuits numbered 1 through 4095. You identify an individual virtual circuit by giving its Logical Channel Identifier (LCI) or Virtual Circuit Number (VCN). Many documents use the terms virtual circuit and LC, VCN, LCN, and LCI interchangeably. Each of these terms refers to the virtual circuit number

An important part of X.25 operation is the range of VCNs. VCNs are broken into four ranges (listed here in numerically increasing order):

1. Permanent virtual circuits (PVCs)
2. Incoming-only circuits
3. Two-way circuits
4. Outgoing-only circuits

The incoming-only, two-way, and outgoing-only ranges define the VCNs over which a Switched Virtual Circuit (SVC) can be established by the placement of an X.25 call, much like a telephone network establishes a switched voice circuit when a call is placed.

The rules about DCE and DTE devices initiating calls are:

- Only the DCE device can initiate a call in the incoming-only range.
- Only the DTE device can initiate a call in the outgoing-only range.
- Both the DCE device and the DTE device can initiate a call in the two-way range.

The ITU-T Recommendation for X.25 defines incoming and outgoing in relation to the DTE or DCE interface role; this documentation uses the more intuitive sense. Unless the ITU-T sense is explicitly referenced, a call received from the interface is an incoming call and a call sent out the interface is an outgoing call.

Note: ***The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).***

There is no difference in the operation of the SVCs ranges except th restrictions on which device can initiate a call. These ranges can be used to prevent one side from monopolizing the virtual circuits, and is useful for X.25 interfaces with a small total number of SVCs available.

Six X.25 parameters define the upper and lower limit of each of th three SVC ranges. A PVC must be assigned a number less than the numbers assigned to the SVC ranges. An SVC range is not allowed to overlap another range.

Note: ***Because the X.25 protocol requires the DTE and DCE to have identical virtual circuit ranges, changes you make to the virtual circuit range limits when the interface is up are held until the X.25 protocol restarts the packet service.***

To configure X.25 virtual circuit ranges, complete these tasks as appropriate for your configuration:

1. Set the lowest incoming-only circuit number (default is 0).

   x25 lic ***circuit-number***

2. Set the highest incoming-only circuit number (default is 0).

   x25 hic ***circuit-number***

3. Set the lowest two-way circuit number (default is 1).

   x25 ltc ***circuit-number***

4. Set the highest two-way circuit number (default is1024 for X.25 and 4095 for CMNS).

   x25 htc ***circuit-number***

5. Set the lowest outgoing-only circuit number (default is 0).

   x25 loc ***circuit-number***

6. Set the highest outgoing-only circuit number (default is 0).

   x25 hoc ***circuit-number***

Each of these parameters can range from 1 to 4095, inclusive. The values for these parameters must be the same on both ends of an X.25 link. For connection to a PDN, these values must be set to the values assigned by the network. Virtual circuit 0 is not available except for marking unused ranges (by setting the lower and upper limits to 0).

Example: Virtual Circuit Ranges

This example sets the virtual circuit ranges of 5 to 20 for incoming calls only (from the DCE to the DTE) and 25 to 1024 for either incoming or outgoing calls. It also specifies no virtual circuits for outgoing calls (from the DTE to the DCE). Up to 4 permanent virtual circuits can be defined on virtual circuits 1 through 4.

```
x25 lic 5
x25 hic 20
x25 ltc 25
```

Set the Packet Numbering Modulo

The AI2524 implementation of X.25 supports both modulo 8 and modulo 128 packet sequence numbering; module 8 is the default.

In interface configuration mode, set the packet numbering modulo:

```
x25 modulo {8 | 128}
```

Note: ***Because the X.25 protocol requires the DTE and DCE to have identical modulos, changes you make to the modulo when the interface is up are held until the X.25 protocol restarts the packet service.***

The X.25 modulo and the LAPB modulo are distinct and serve different purposes. LAPB modulo 128 (or extended mode) can be used to achieve higher throughput across the DTE or DCE interface; it affects only the local point of attachment. X.25 Packet-Level Protocol (PLP) modulo 128 can be used to achieve higher end-to-end throughput for virtual circuits by allowing more data packets to be in transit through the X.25 network.

Set the X.121 Address

If your router does not originate or terminate calls but only participates in X.25 switching, this task is optional. However, if the router is at tached to a PDN, you must set the interface X.121 address assigned by the X.25 network service provider.

In interface configuration mode, set the X.121 address:

```
x25 address x121-address
```

Set the Default Flow Control Values

Because X.25 is a strongly flow-controlled protocol, setting correct default flow control parameters for window size and packet size is essential. Mismatched default flow control values will cause X.25 local procedure errors, evidenced by Clear and Reset events.

To configure flow control parameters, complete these tasks. Thes tasks are optional if your X.25 attachment uses the standard default values for maximum packet sizes (128 bytes incoming and outgoing and window sizes (2 packets incoming and outgoing).

- Set Default Window Sizes
- Set Default Packet Sizes

Note: ***Because the X.25 protocol requires the DTE and DCE to have identical default maximum packet sizes and default window sizes, changes made to the window and packet sizes when the interface is up are held until the X.25 protocol restarts the packet service.***

Set Default Window Sizes

X.25 networks have a default input and output window size (the preset value is 2) that is defined by your network administrator. You must set the Cisco IOS software default input and output window sizes to match those of the network. These defaults are the values that an SVC uses if it is set up without explicitly negotiating its window sizes. PVCs also use these default values unless different values are configured.

In interface configuration mode, set the default window sizes:

1. Set the default virtual circuit receive window size.

   ```
   x25 win packets
   ```

2. Set the default virtual circuit transmit window size.

   ```
   x25 wout packets
   ```

Set Default Packet Sizes

X.25 networks have a default maximum input and output packet siz (the preset value is 128) that is defined by your network administrator. You must set the Cisco IOS software default input and output maximum packet sizes to match those of the network. These defaults are the values that an SVC uses if it is set up without explicitly negotiating its maximum packet sizes. PVCs also use these default values unless different values are configured.

In interface configuration mode, set the default input and output maximum packet sizes:

1. Set the default input maximum packet size.

```
x25 ips bytes
```

2. Set the default output maximum packet size.

```
x25 ops bytes
```

To send a packet larger than the agreed on X.25 packet size over an X.25 virtual circuit, the Cisco IOS software must break the packet into two or more X.25 packets with the M-bit (More data bit) set. The receiving device collects all packets in the M-bit sequence and reassembles them into the original packet.

It is possible to define default packet sizes that cannot be supported by the lower layer (see the LAPB N1 parameter). However, the router will negotiate lower maximum packet sizes for all SVCs so the agreed on sizes can be carried. The Cisco IOS software will also refuse a PVC configuration if the resulting maximum packet sizes cannot be supported by the lower layer.

Example: Typical X.25 Configuration

This example shows the complete configuration for a serial interface connected to a commercial X.25 PDN for routing the IP protocol. The IP subnetwork address 172.25.9.0 has been assigned for the X.25 network.

Note: ***When you are routing IP over X.25, you must treat the X.25 network as a single IP network or subnetwork. Map entries for routers with addresses on subnetworks other than the one on which the interface's IP address is stored are ignored by the routing software. All routers using the subnet number must have map entries for all others routers. Using the broadcast option with dynamic routing can result in larger traffic loads, requiring a larger hold queue, larger window sizes, or multiple virtual circuits.***

```
interface serial 2
ip address 172.25.9.1 255.255.255.0
!
encapsulation X25
!
```

The bandwidth command is not part of the X.25 configuration; it is especially important to understand that it does not have any connection with the X.25 entity of the same name. bandwidth commands are used by IP routing processes (currently only IGRP) to determine which lines are the best choices for traffic. Since the default is 1544 Kbaud, and X.25 service at that rate is not generally available, most X.25 in terfaces that are being used with IGRP in a real environment will have bandwidth settings.

This is a 9.6 Kbaud line:

```
bandwidth 10
```

These Level 3 parameters are default flow control values; they need to match the PDN defaults. The values used by an SVC are negotiable on a per-call basis:

```
x25 win 7
x25 wout 7
x25 ips 512
x25 ops 512
```

You must specify an X.121 address to be assigned to the X.25 interface by the PDN.

```
x25 address 31370054065
```

These Level 3 parameters have been set to match the network. You generally need to change some Level 3 parameters, most often those listed below. You might not need to change any Level 2 parameters, however.

```
x25 htc 32
x25 idle 5
x25 nvc 2
```

These commands configure the X.25 map. If you want to exchange routing updates with any of the routers, they would need broadcast flags. If the X.25 network is the only path to the routers, static routes are generally used to save on packet charges. If there is a redundant path, it might be desirable to run a dynamic routing protocol.

```
x25 map IP 172.25.9.3 31370019134 ACCEPT-REVERSE
x25 map IP 172.25.9.2 31370053087
```

ACCEPT-REVERSE allows collect calls.

If the PDN cannot handle fast back-to-back frames, use the **transmitter-delay** command to slow down the interface.

```
transmitter-delay 1000
```

Configure Additional X.25 Interface Parameters

Some X.25 applications have less common, or special, needs. Several X.25 parameters are available to modify the X.25 protocol behavior for these applications.

To configure less common X.25 interface parameters for these special needs, perform these tasks, as needed:

- Configure the X.25 Level 3 Timers
- Configure X.25 Addresses
- Establish a Default Virtual Circuit Protocoll
- Disable Packet-Level Protocol (PLP) Restarts

Configure the X.25 Level 3 Timers

The X.25 Level 3 retransmission timers determine how long the Cisco IOS software waits for acknowledgment of control packets. You can set these timers independently. Only those timers that apply to the interface are able to be configured. (A DTE interface does not have the T1x timers, and a DCE interface does not have the T2x timers.)

To set the retransmission timers, perform any of these tasks in interface configuration mode:

- Set DTE T20 Restart Request.

  x25 t20 ***seconds***

- Set DCE T10 Restart Indication.

  x25 t10 ***seconds***

- Set DTE T21 Call Request.

  x25 t21 ***seconds***

- Set DCE T11 Incoming Call.

  x25 t11 ***seconds***

- Set DTE T22 Reset Request.

  x25 t22 ***seconds***

- Set DCE T12 Reset Indication.

  x25 t12 ***seconds***

- Set DTE T23 Clear Request.

  x25 t23 ***seconds***

- Set DCE T13 Clear Indication.

  x25 t13 ***seconds***

Configure X.25 Addresses

When establishing SVCs, X.25 uses addresses in the form defined by the ITU-T Recommendation for X.121. An X.121 address has from zero to 15 digits. Because of the importance of addressing to call setup, several interface addressing features are available for X.25.

To configure X.25 addresses, perform these tasks:

- Understand Normal X.25 Addressing
- Understand X.25 Subaddresses
- Configure an Interface Alias Address
- Suppress or Replace the Calling Address
- Suppress the Called Address

Understand Normal X.25 Addressing

An X.25 interface's X.121 address is used when it is the source or destination of an X.25 call. The X.25 call setup procedure identifies both the calling (source) and the called (destination) X.121 addresses. When an interface is the source of a call, it encodes the interface X.121 address as the source address. An interface determines that it is th destination of a received call if the destination address matches the interface's address.

The AI2524 X.25 software can also route X.25 calls, which involves placing and accepting calls, but the router is neither the source nor the destination for these calls. Routing X.25 does not modify the source or destination addresses, thus preserving the addresses specified by the source host. Routed (switched) X.25 simply connects two logical X.25 channels to complete an X.25 virtual circuit. An X.25 virtual circuit, is a connection between two hosts (the source host and the destination host) that is switched between routed X.25 links.

The null X.121 address (the X.121 address that has zero digits) is a special case. The router acts as the destination host for any call it receives that has the null destination address.

Understand X.25 Subaddresses

A subaddress is an X.121 address that matches the digits defined for the interface's X.121 address, but has additional digit(s) after the base address. X.25 acts as the destination host for an incoming Packet Assembler/Disassembler (PAD) call with a destination that is a subaddress of the interface's address; the trailing digits specify which line a PAD connection is requesting. Other calls that use a subaddress can be accepted if the trailing digit(s) are zeros; otherwise, the router will not act as the call's destination host.

Configure an Interface Alias Address

You can supply alias X.121 addresses for an interface. This allows the interface to act as the destination host for calls having a destination address that is neither the interface's address, an allowed subaddress o the interface, nor the null address.

Local processing (for example, IP encapsulation) can be performed only for incoming calls whose destination X.121 address matches the serial interface or alias of the interface.

In global configuration mode, configure an alias:

```
x25 route [#position] x121-address-pattern
[cud pattern] alias type number
```

Suppress or Replace the Calling Address

Some attachments require that no calling (source) address be presented in outgoing calls. The requirement is called suppressing the calling address.

When attached to a PDN, X.25 may need to ensure that outgoing calls only use the assigned X.121 address for the calling (source) address. Routed X.25 normally uses the original source address. Although individual X.25 route configurations can modify the source address, Cisco provides a simple command to force the use of the interface address in all calls sent; this requirement is called replacing the calling address.

To suppress or replace the calling address, perform the appropriat task in interface configuration mode:

- Suppress the calling (source) X.121 address in outgoing calls.

  ```
  x25 suppress-calling-address
  ```

- Replace the calling (source) X.121 address in switched calls.

  ```
  x25 use-source-address
  ```

Suppress the Called Address

Some attachments require that no called (destination) address be presented in outgoing calls; this requirement is called suppressing the called address.

In interface configuration mode, suppress the called address:

```
x25 suppress-called-address
```

Establish a Default Virtual Circuit Protocol

The Call Request packet that sets up a virtual circuit can encode a field called the Call User Data (CUD) field. Typically, the first few bytes of the CUD field identify which high-level protocol is carried by the virtual circuit. The router, when acting as a destination host, normally refuses a call if the CUD is absent or if the protocol identification isn't recognized. The PAD protocol, however, specifies that unidentified calls be treated as PAD connection requests. Other applications require that they be treated as IP encapsulation connection requests, per RFC 877.

In interface configuration mode, configure either PAD or IP encapsu lation treatment of unidentified calls:

```
x25 default {ip | pad}
```

Disable Packet-Level Protocol (PLP) Restarts

By default, a PLP restart is performed when the link level resets (for example, when LAPB reconnects). Although PLP restarts can be disabled for those few networks that do not allow restarts, do not disable these restarts because doing so can cause anomalous packet layer behavior.

In interface configuration mode, disable PLP restarts:

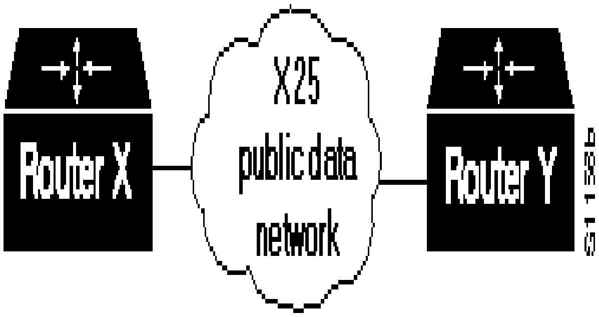
```
no x25 linkrestart
```

Configure an X.25 Datagram Transport

X.25 support is most commonly configured as a transport for data grams across an X.25 network. Datagram transport (or encapsulation) is a cooperative effort between two hosts communicating across an X.25 network. You configure datagram transport by establishing a mapping on the encapsulating interface between the far host's protocol address (for example, IP) and its X.121 address. Because the call identifies the protocol that the virtual circuit will carry (in the CUD field), the terminating host can accept the call if it is configured to exchange the identified traffic with the source host.

Figure 12-1 illustrates two routers sending datagrams across an X.25 public data network (PDN).

Figure 12-1: Transporting LAN Protocols across an X.25 Public Data Network (PDN)



Perform these tasks, as necessary, to complete the X.25 configuration for your network needs:

- Configure Subinterfaces
- Map Protocol Addresses to X.121 Addresses
- Establish an Encapsulation PVC
- Set X.25 TCP/IP Header Compression
- Configure X.25 Bridging

These sections describe how to perform these configuration tasks.

Configure Subinterfaces

Subinterfaces are virtual interfaces that can be used to connect several networks to each other through a single physical interface. Subinterfaces are made available on AI2524 routers because routing protocols, especially those using the split horizon principle, may need help to determine which hosts need a routing update. The split horizon principle allows routing updates to be distributed to other routed interfaces ex cept the interface on which the routing update was received. It works well in a LAN environment in which other routers reached by the interface have already received the routing update.

However, in a WAN environment using connection-oriented interfaces (like X.25 and Frame Relay), other routers reached by the same physical interface might not have received the routing update. Rathe than forcing you to connect routers by separate physical interfaces, the AI2524 provides subinterfaces that are treated as separate interfaces. You can separate hosts into subinterfaces on a physical interface. Separation does not affect the X.25 protocol, and routing processes recognize each subinterface as a separate source of routing updates, enabling all subinterfaces to receive routing updates.

Understand Point-to-Point and Multipoint Subinterfaces

There are two types of subinterfaces: point-to-point and multipoint. Subinterfaces are implicitly multipoint unless configured as point-to-point.

A point-to-point subinterface is used to encapsulate protocols between two hosts. An X.25 point-to-point subinterface will accept only a single encapsulation command (such as **`x25 map`** or **`x25 pvc`**) for a given protocol, so there can be only one destination for the protocol. However, you can use multiple encapsulation commands, one for each protocol, or multiple protocols for one map or PVC. All protocol traffic routed to a point-to-point subinterface is forwarded to the one destination host defined for the protocol. Because only one destination is defined for the interface, the routing process need not consult the destination address in the datagrams.

A multipoint subinterface is used to connect hosts for a given protocol. There is no restriction on the number of encapsulation commands that can be configured on a multipoint subinterface. Because the hosts appear on the same subinterface, they are not relying on the router to distribute routing updates between them. When a routing process forwards a datagram to a multipoint subinterface, the X.25 encapsulation process must be able to map the datagram's destination address to a configured encapsulation command. If the routing process cannot find a map for the datagram destination address, the encapsulation will fail.

Note: ***Because of the complex operations dependent on a subinterface and its type, the router will not allow a subinterface type to be changed, nor can a subinterface with the same number be re-established once it has been deleted. After a subinterface has been deleted, you must reload the Cisco IOS software (by using the*** **`reload`** ***command) to remove all internal references. However, you can easily reconstitute the deleted subinterface by using a different subinterface number.***

Create and Configure X.25 Subinterfaces

In interface configuration mode, create and configure a subinterface by completing the first task and one or both of the second tasks:

1. Create a point-to-point or multipoint subinterface.

   ```
   interface serial number.subinterface-number
   [point-to-point | multipoint]
   ```

2. Configure an X.25 encapsulation map for the subinterface:

   ```
   x25 map protocol address [protocol2
   address2 [... [protocol9 address9]]] x121-
   address [option]
   ```

And/or establish an encapsulation PVC for the subinterface:

```
x25 pvc circuit protocol address [protocol2
address2 [...[protocol9 address9]]] x121-
address [option]
```

Note: ***When configuring IP routing over X.25, you might need to make adjustments to accommodate split horizon effects. Refer to the Configuring IP Routing Protocols chapter in the Network Protocols Configuration Guide, Part 1 for details about how the Cisco IOS software handles possible split horizon conflicts. By default, split horizon is enabled for X.25 networks.***

Example: Point-to-Point Subinterface Configuration

This example creates a point-to-point subinterface and maps IP to a remote host:

```
interface Serial0.1 point-to-point
x25 map ip 172.20.170.90 170090 broadcast
```

Map Protocol Addresses to X.121 Addresses

This section describes the X.25 single-protocol and multiprotocol encapsulation options that are available and describes how to map protocol addresses to an X.121 address for a remote host. This section also includes reference information about how protocols are identified.

Understand Protocol Encapsulation for Single-Protocol and Multiprotocol Virtual Circuits

The AI2524 supports encapsulation of a number of datagram protocols across X.25, using a standard method when available, or a proprietary method when necessary. These traditional methods assign a protocol to each virtual circuit. If more than one protocol is carried between th router and a given host, each active protocol will have at least one virtual circuit dedicated to carrying its datagrams.

The AI2524 also supports RFC 1356, a standardized method for en capsulating most datagram protocols over X.25. It also specifies ho one virtual circuit can carry datagrams from more than one protocol.

The Cisco IOS software can be configured to use any of the available encapsulation methods with a particular host.

After you establish an encapsulation virtual circuit, the Cisco IOS software sends and receives a datagram by simply fragmenting it from and reassembling it into an X.25 complete packet sequence. An X.25 complete packet sequence is one or more X.25 data packets that have the M-bit set in all but the last packet. A virtual circuit that can carry multiple protocols includes protocol identification data as well as the protocol data at the start of each complete packet sequence.

Understand Protocol Identification

The various methods and protocols used in X.25 SVC encapsulation are identified in a specific field of the call packet; this field is defined by X.25 to carry CUD. Since PVCs do not use the X.25 call setup procedures, only PVCs do not use CUD to identify their encapsulation.

The primary difference between the available AI2524 and IETF encapsulation methods is the specific value used to identify a protocol. When any of the methods establishes a virtual circuit for carrying single protocol, the protocol is identified in the call packet by the CUD. When a virtual circuit is established to carry more than one protocol (only available using the RFC 1356 methodology), a protocol identification field precedes the datagram encapsulated in the X.25 data packet; every datagram exchanged over that virtual circuit has its protocol identified.

This table summarizes the values used in the CUD field to identify protocols.

Protocol	AI2524 Protocol Identifier	IETF RFC 1356 Protocol Identifier
Apollo Domain	0xD4	0x80 (5-byte SNAP encoding[1])
Bridging	0xD5	(Not implemented)
ISO CLNS	0x81	0x81[2]
Compressed TCP	0xD8	0x00 (5-byte SNAP encoding)[3]
IP	0xCC	0xCC[4] or 0x80 (5-byte SNAP encoding)
Novell IPX	0xD3	0x80 (5-byte SNAP encoding)
PAD	0x01[5]	0x01[5]
QLLC	0xC3	(Not available)
Multiprotocol	(Not available)	0x00

1. Subnetwork Access Protocol (SNAP) encoding is defined from the Assigned Numbers RFC. The AI2524 implementation recognizes only the IETF organizational unique identifier (OUI) 0x0000 00 followed by a 2-byte Ethernet protocol type.

2. The use of 0x81 for CLNS is compatible with ISO/IEC 8473-3:1994.

3. Compressed TCP traffic has two types of datagrams, so IETF encapsulation requires a multiprotocol virtual circuit.

4. The use of 0xCC for IP is backwards-compatible with RFC 877.

5. The use of 0x01 for PAD is defined by ITU-T Recommendation X.29.

Once a multiprotocol virtual circuit has been established, datagrams on the virtual circuit have protocol identification data before the actual protocol data; the protocol identification values are the same used by RFC 1356 in the CUD field for an individual protocol.

Note: ***IP datagrams can be identified with a 1-byte identification (0xCC) or a 6-byte identification (0x80 followed by the 5-byte SNAP encoding). The 1-byte encoding is used b default, although the SNAP encoding can be configured.***

Map Datagram Addresses to X.25 Hosts

Encapsulation is a cooperative process between the router and anothe X.25 host. Because X.25 hosts are reached with an X.121 address (an X.121 address has up to15 decimal digits), the router must have means to map a host's protocols and addresses to its X.121 address.

Each encapsulating X.25 interface must be configured with the relevant datagram parameters. For example, an interface that encapsulates IP will typically have an IP address.

You must also establish the X.121 address of an encapsulating X.25 interface using the **`x25 address`** interface configuration command. The X.121 address is the address where encapsulation calls ar directed. This is also the source X.121 address used for originating an encapsulation call and is used by the destination host to map the sourc host and protocol to the protocol address. An encapsulation virtual circuit must be a mapped at both the source and destination host inter faces.

For each X.25 interface, you must explicitly map each destination host's protocols and addresses to its X.121 address. If needed and if the destination host has the capability, one host map can be configured to support several protocols; alternatively, you can define one map for each supported protocol.

In interface configuration mode, establish a map:

x25 map ***protocol address*** **[*protocol2 address2*[...[*protocol9 address9*]]]** ***x121-address*** **[*option*]**

For example, if you are encapsulating IP over a given X.25 interface, you must define an IP address for the interface and, for each of the desired destination hosts, map the host's IP address to its X.121 address.

Note: ***You can map an X.121 address to as many as nine protocol addresses, but each protocol can be mapped only once in the command line.***

An individual host map can use these keywords to specify these protocols:

- **apollo**—Apollo Domain
- **bridge**—Bridging
- **clns**—OSI Connectionless Network Servic
- **compressedtcp**—TCP/IP header compression
- **ip**—IP
- **ipx**—Novell IPX
- **pad**—Packet Assembler/Disassemble
- **qllc**—IBM's QLLC

Each mapped protocol, except bridging and CLNS, takes a datagram address. All bridged datagrams are sent to all bridge maps on an interface. CLNS uses the mapped X.121 address as the SNPA, which is referenced by a **clns neighbor** command. The configured datagram protocol(s) and relevant addresses are mapped to the destination host's X.121 address. All protocols that are supported for RFC 1356 opera tion can be specified in a single map. Bridging and QLLC are not supported for RFC 1356 encapsulation. If IP and TCP/IP header compression are both specified, the same IP address must be given for both protocols.

When setting up the address map, you can include options such as enabling broadcasts, specifying the number of virtual circuits allowed, and defining various user facility settings.

Note: ***Multiprotocol maps, especially those configured to carr broadcast traffic, can result in significantly larger traffi loads, requiring a larger hold queue, larger window sizes, or multiple virtual circuits.***

You can simplify the configuration for the Open Shortest Path First (OSPF) protocol by adding the optional **`broadcast`** keyword.

Configure PAD Access

By default, PAD connection attempts are processed for session cre ation or protocol translation from all hosts. In interface configuration mode, restrict PAD connections to only statically mapped X.25 hosts:

1. Restrict PAD access.

   ```
   x25 pad-access
   ```

2. Configure a host for PAD access.

   ```
   x25 map pad x121-address [option]
   ```

You can configure outgoing PAD access using the optional features of the **`x25 map pad`** command without restricting incoming PAD connections to the configured hosts.

Establish an Encapsulation PVC

Permanent Virtual Circuits (PVCs) are the X.25 equivalent of leased lines; they are never disconnected. You do not need to configure an address map before defining a PVC; an encapsulation PVC implicitly defines a map.

In interface configuration mode, establish a PVC:

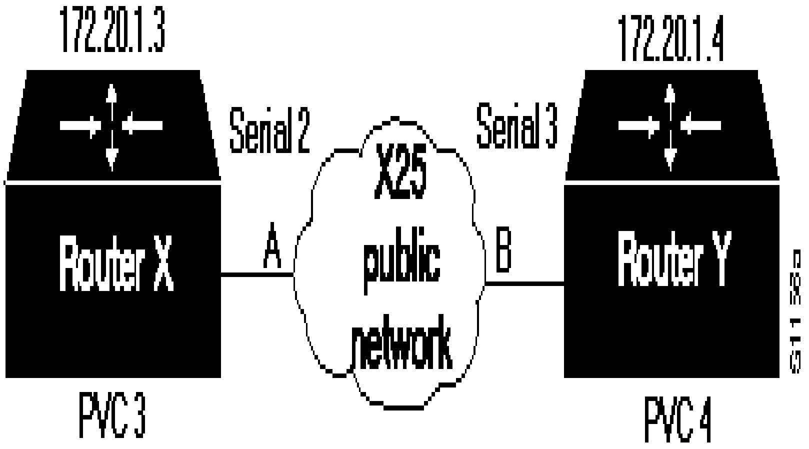
```
x25 pvc circuit protocol address [protocol2
address2 [...[protocol9 address9]]] x121-
address [option]
```

The **`x25 pvc`** command uses the same protocol keywords as the **`x25 map`** command. Encapsulation PVCs also use a subset of the options defined for the **`x25 map`** command.

Example: PVC Used to Exchange IP Traffic

This example, illustrated in Figure 12-2, demonstrates how to use the PVC to exchange IP traffic between Router X and Router Y.

Figure 12-2:Establishing an IP Encapsulation PVC through an X.25 Network



Configuration for Router X

```
interface serial 2
ip address 172.20.1.3 255.255.255.0
x25 map ip 172.20.1.4 0
x25 pvc 4 ip 172.20.1.4
```

Configuration for Router Y

```
interface serial 3
ip address 172.20.1.4 255.255.255.0
x25 map ip 172.20.1.3 0
x25 pvc 3 ip 172.20.1.3
```

In this example, the PDN has established a PVC through its network connecting PVC number 3 of access point A to PVC number 4 of access point B. On Router X, a connection is established between Router X and Router Y's IP address, 172.20.1.4. On Router Y, a connection is established between Router Y and Router X's IP address, 172.20.1.3.

Set X.25 TCP/IP Header Compression

The AI2524 supports RFC 1144 TCP/IP header compression (THC) on serial lines using HDLC and X.25 encapsulation. THC encapsulation is different from other encapsulation traffic. The implementation of compressed TCP over X.25 uses one virtual circuit to pass the compressed packets. Any IP traffic (including standard TCP) is separate from TCH traffic; it is carried over separate IP encapsulation virtual circuits or identified separately in a multiprotocol virtual circuit.

Note: *If you specify both* **`ip`** *and* **`compressedtcp`** *in the same* **`x25 map compressedtcp`** *command, they must both specify the same IP address.*

In interface configuration mode, set up a separate virtual circuit for X.25 TCP/IP header compression:

```
x25 map compressedtcp ip-address [protocol2
address2 [...[protocol9 address9 ]]] x121-
address [option]
```

Configure X.25 Bridging

The AI2524 transparent bridging software supports bridging ove X.25 virtual circuits. Bridging is not supported for RFC 1356 operation. Bridge maps must include the broadcast option for correct operation.

In interface configuration mode, enable the X.25 bridging capability:

```
x25 map bridge x121-address broadcast
[option]
```

Configure Additional X.25 Datagram Transport Features

The Cisco IOS software allows you to configure additional X.25 datagram transport features, including various user facilities defined for X.25 call setup.

This section describes the X.25 datagram transport features you can configure by using the options in the ***x25 map*** or **x25 pvc encapsulation** command (or by setting an interface default). Th tasks you perform depend upon your needs, the structure of your network, and the requirements of the service provider.

To configure the optional parameters, user facilities, and special features, perform one or more of these tasks:

- Configure X.25 Payload Compression
- Configure the Encapsulation Virtual Circuit Idle Time
- Increase the Number of Virtual Circuits Allowed
- Configure the Ignore Destination Time
- Establish the Packet Acknowledgment Policy
- Configure X.25 User Facilities
- Define the Virtual Circuit Packet Hold Queue Size
- Restrict Map Usage

Configure X.25 Payload Compression

For increased efficiency on relatively slow networks, the Cisco IOS software supports X.25 payload compression of outgoing encapsulation traffic.

Several restrictions apply to X.25 payload compression:

- The compressed virtual circuit must connect two Cisco routers, because X.25 payload compression is not standardized.

  The data packets conform to the X.25 protocol rules, so a compressed virtual circuit can be switched through standard X.25 equipment. However, only Cisco routers can compress and decompress the data.

- Only datagram traffic can be compressed, although all the encapsulation methods supported by Cisco routers are available. For example, an IETF multiprotocol virtual circuit can be compressed.

  SVCs cannot be translated between compressed and uncom pressed data, nor can PAD data be compressed.

- X.25 payload compression must be applied carefully.

  Each compressed virtual circuit requires significant memory resources (for a dictionary of learned data patterns) and computation resources (every data packet received is decompressed and every data packet sent is compressed). Excessive use of compression can cause unacceptable overall performance.

- X.25 compression must be explicitly configured for **`map`** command.

  A received call that specifies compression will be rejected if the corresponding host map does not specify the compress option. An incoming call that does not specify compression can, however, be accepted by a map that specifies compression.

In interface configuration mode, enable payload compression over X.25:

```
x25 map protocol address [protocol2
address2 [...[protocol9 address9]]] x121-
address compress
```

This command specifies that X.25 compression is to be used between the two hosts. Because each virtual circuit established for compressed traffic uses significant amounts of memory, compression should be used with careful consideration of its impact on performance.

The **`compress`** option may be specified for an encapsulation PVC.

Configure the Encapsulation Virtual Circuit Idle Time

The Cisco IOS software can clear a datagram transport SVC after a set period of inactivity. Routed SVCs are not timed for inactivity.

In interface configuration mode, set the time:

1. Set an idle time for clearing encapsulation.

   ```
   x25 idle minutes
   ```

2. Specify an idle time for clearing a map's SVCs.

   ```
   x25 map protocol address [protocol2
   address2 [...[protocol9 address9]]] x121-
   address idle minutes
   ```

Increase the Number of Virtual Circuits Allowed

For X.25 datagram transport, you can establish up to eight SVCs to one host for each map.

To increase the number of virtual circuits allowed, perform one or both of these tasks in interface configuration mode:

- Specify the default maximum number of SVCs that can be open simultaneously to one host for each map.

  x25 nvc *count*

- Specify the maximum number of SVCs allowed for a map.

  x25 map *protocol address* [*protocol2 address2* [...[*protocol9 address9*]]] *x121-address* nvc *count*

Configure the Ignore Destination Time

Upon receiving a Clear Request for an outstanding datagram transport Call Request, the X.25 encapsulation code immediately tries another Call Request if it has more traffic to send. This action can overrun some X.25 switches.

To define the number of minutes the Cisco IOS software will prevent calls from going to a previously failed destination, type this command in interface configuration mode. Incoming calls will still be accepted.

x25 hold-vc-timer *minutes*

Establish the Packet Acknowledgment Policy

You can instruct the Cisco IOS software to send an acknowledgment packet when it has received a threshold of data packets it has not acknowledged, instead of waiting until its input window is full. A value of 1 sends an acknowledgment for each data packet received if it cannot be acknowledged in an outgoing data packet. This approach improves line responsiveness at the expense of bandwidth. A value of 0 restores the default behavior of waiting until the input window is full.

To establish the acknowledgment threshold, type (in interface configuration mode):

x25 th *delay-count*

The packet acknowledgment threshold also applies to encapsulation PVCs.

Configure X.25 User Facilities

The X.25 software provides commands to support X.25 user facilities (options specified by the creators of the X.25 Recommendation) that allow you to implement features such as accounting, user identification, and flow control negotiation. You can choose to configure facilities on a per-map basis or on a per-interface basis. In the following list, the **x25 map** commands configure facilities on a per-map basis; the **x25 facility** commands specify the values sent for all encapsulation calls originated by the interface. Routed calls are not affected by the facilities specified for the outgoing interface.

To set the supported X.25 user facilities, perform one or more of these tasks in interface configuration mode:

- Select the closed user group.

  x25 facility cug ***group-number***

  or

  x25 map ***protocol address*** **[*protocol2 address2* [...[*protocol9 address9*]]]** ***x121-address*** **cug** ***number***

- Set flow control parameter negotiation values to request on outgoing calls.

  x25 facility packetsize ***in-size out-size***

  or

  x25 map ***protocol address*** **[*protocol2 address2* [...[*protocol9 address9*]]]** ***x121-address*** **packetsize** ***in-size out-size***

  x25 facility windowsize ***in-size out-size***

  or

  x25 map ***protocol address*** **[*protocol2 address2* [...[*protocol9 address9*]]]** ***x121-address*** **windowsize** ***in-size outsize***

- Set reverse charging.

  x25 facility reverse

  or

  x25 map ***protocol address*** **[*protocol2 address2* [...[*protocol9 address9*]]]** ***x121-address*** **reverse**

- Allow reverse charging acceptance.

  ```
  x25 accept-reverse
  ```

  or

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address accept-reverse
  ```

- Select throughput class negotiation.

  ```
  x25 facility throughput  in out
  ```

  or

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address throughput  in out
  ```

- Select transit delay.

  ```
  x25 facility transit-delay  value
  ```

  or

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address transit-delay  milliseconds
  ```

- Set the Recognized Private Operation Agency (RPOA) to use.

  ```
  x25 facility rpoa  name
  ```

  or

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address rpoa  name
  ```

- Set the AI2524 standard network user identification.

  ```
  x25 map protocol address [protocol2
  address2 [...[ protocol9 address9]]] x121-
  address nuid username password
  ```

- Set a user-defined network user identification allowing the format to be determined by your network administrator.

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address nudata string
  ```

The **windowsize** and **packetsize** options are supported for PVCs, although they have a slightly different meaning because PVCs do not use the call setup procedure. If the PVC does not use the interface defaults for the flow control parameters, these options must b used to specify the values. Not all networks will allow a PVC to be defined with arbitrary flow control values.

Additionally, the Data bit (D-bit) is supported, if negotiated. PVCs allow the D-bit procedure because there is no call setup to negotiate its use. Both restricted and unrestricted fast select are also supported and are transparently handled by the software. No configuration is required for use of the D-bit or fast select facilities.

Define the Virtual Circuit Packet Hold Queue Size

To define the maximum number of packets that can be held whil virtual circuit is unable to send data, type (in interface configuration mode):

```
x25 hold-queue queue-size
```

An encapsulation virtual circuit's hold queue size is determined when it is created; the **x25 hold-queue** command does not affect existing virtual circuits. This command also defines the hold queue siz of encapsulation PVCs.

Restrict Map Usage

An X.25 map can be restricted so that it will not be used to place calls or so that it will not be considered when incoming calls are mapped.

To restrict X.25 map usage, use the following map options as needed:

- Restrict incoming calls from a map.

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address no-incoming
  ```

- Restrict outgoing calls from a map.

  ```
  x25 map protocol address [protocol2
  address2 [...[protocol9 address9]]] x121-
  address no-outgoing
  ```

Configure X.25 Routing

The X.25 software implementation allows virtual circuits to be routed from one X.25 interface to another and from one router to another. The routing behavior can be controlled with switching and X.25-over-TCP (XOT) configuration commands, based on a locally built table.

X.25 encapsulation can share an X.25 serial interface with the X.25 switching support. Switching or forwarding of X.25 virtual circuits can be completed two ways:

- Incoming calls received from a local serial interface running X.25 can be forwarded to another local serial interface running X.25. This is known as local X.25 switching because the router handles the complete path. It does not matter whether the interfaces are configured as DTE or DCE device because the software takes the appropriate actions.
- An incoming call also can be forwarded to another Cisco router over a LAN using the TCP/IP protocols. Upon receipt of an incoming call, a TCP connection is established to the router that is acting as the switch for the destination. All X.25 packets are sent and received over this reliable data stream. Flow control is maintained end-to-end. This is known as X.25-over-TCP or XOT. (XOT was previously called remote switching or tunneling.) It does not matter whether the interfaces are configured as DTE or DCE, because the software takes the appropriate actions.

Running X.25 over TCP/IP provides a number of benefits. The datagram containing the X.25 packet can be switched by other routers using their high-speed switching abilities. X.25 connections can be sent over networks running only the TCP/IP protocols. The TCP/IP protocol suite runs over many different networking technologies, including Ethernet, Token Ring, T1 serial, and FDDI. Thus X.25 data can be forwarded over these media to another router where it can b output to an X.25 interface.

When the connection is made locally, the switching configuration is used; when the connection is across a LAN, the XOT configuration is used. The basic function is the same for both types of connections, but different configuration commands are required for each type of connection.

The X.25 switching subsystem supports these facilities and parame ters:

- D-bit negotiation—Data packets with the D-bit set are passed through transparently
- Variable-length interrupt data
- Flow control parameter negotiation
- Window size up to 7, or 127 for modulo 128 operation
- Packet size up to 4096, if the LAPB layers used are capable of handling the requested siz
- Basic closed user group selection
- Throughput class negotiation
- Reverse charging and fast select

To configure X.25 routing, perform these tasks:

- Enable X.25 Routing
- Configure a Local X.25 Route
- Configure XOT (Remote) X.25 Rout
- Configure a Locally Switched PVC
- Configure an XOT (Remote) PVC

You may also need to configure additional X.25 routing features, as required for your network.

Enable X.25 Routing

In global configuration mode, enable X.25 routing to use local switching or XOT by typing:

```
x25 routing [use-tcp-if-defs]
```

The **use-tcp-if-defs** keyword is used by some routers that re ceive remote routed calls from older versions of XOT; it might be needed if the originating router cannot be migrated to a new softwar release.

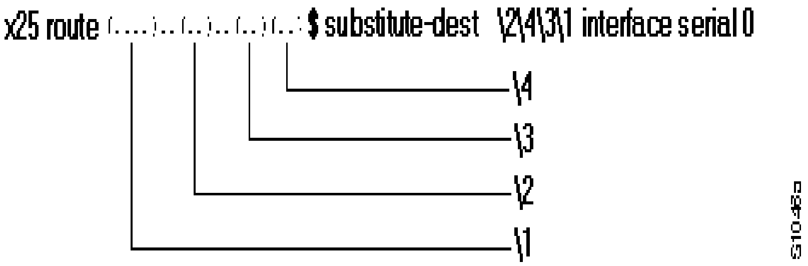
Example: X.25 Route Address Pattern Matching

This example shows how to route X.25 calls with addresses whos first four Data Network Identification Code (DNIC) digits are 1111 to interface serial 3, and to change the DNIC field in the addresses pre sented to the equipment connected to that interface to 2222. The \1 in the rewrite pattern indicates the portion of the original address matched by the digits following the 1111 DNIC.

```
x25 route ^1111(.*) substitute-dest 2222\1 interface serial 3
```

Figure 12-3 shows a more contrived command intended to illustrate the power of the rewriting scheme.

Figure 12-3:X.25 Route Address Pattern Matching


x25 route (....)..(..)..(..)(..)$ substitute-dest \2\4\3\1 interface serial 0
\4
\3
\2
\1
S1046a

The command in Figure 12-3 causes all X.25 calls with 14-digit called addresses to be routed through interface serial 0. The incoming DNIC field is moved to the end of the address. The fifth, sixth, ninth, and tenth digits are deleted, and the thirteenth and fourteenth are moved before the eleventh and twelfth.

Configure a Local X.25 Route

When an incoming call needs to be forwarded, two fields in the X.25 routing table are consulted to determine a local X.25 route:

- Destination X.121 address
- X.25 packet's CUD field (optional)

When the destination address and the CUD of the incoming packet fit the X.121 and CUD patterns in the routing table, the call is forwarded. Forwarding to a specified interface is called local routing or local switching.

To configure a local X.25 route, thus adding the local route to the routing table, type (in global configuration mode):

```
x25 route [#position] x121-address [cud
pattern] interface type number
```

Example: X.25 Routing

This example shows how to enable X.25 switching, as well as how to enter routes into the X.25 routing table:

Enable X.25 forwarding:

```
x25 routing
```

Enter routes into the table. Without a positional parameter, entries ar appended to the end of the table.

```
x25 route ^100$ interface serial 0
x25 route 100 cud ^pad$ interface serial 2
x25 route 100 interface serial 1
x25 route ^3306 interface serial 3
x25 route .* ip 10.2.0.2
```

The routing table forwards calls for X.121 address 100 out interface serial 0. Otherwise, calls are forwarded onto serial 1 if the X.121 address contains 100 anywhere within it and contains no CUD. Also, if the CUD is not the string pad, calls are forwarded to serial 1. If the X.121 address contains the digits 100 and the CUD is the string pad, the call is forwarded onto serial 2. All X.121 addresses that do not match the first 3 routes are checked for a DNIC of 3306 as the first 4 digits. If they do match, they are forwarded over serial 3. All other X.121 addresses will match the fifth entry, which is a match-all pattern and will have a TCP connection established to the IP address 10.2.0.2. The router at 10.2.0.2 will then route the call according to its X.25 routing table.

This example configures a router that sits on a Tymnet/PAD switch to accept calls and have them forwarded to a DEC VAX system. This feature permits running an X.25 network over a generalized existing IP network, thereby making another physical line for one protocol unnecessary. The router positioned next to the DEC VAX system is configured with X.25 routes.

```
x25 route vax-x121-address interface serial 0
x25 route .* ip cisco-on-tymnet-ipaddress
```

These commands route all calls to the DEC VAX X.121 address out to serial 0, where the VAX is connected running PSI. All other X.121 addresses are forwarded to the cisco-on-tymnet address through its IP address. As a result, all outgoing calls from the VAX are sent to cisco on-tymnet for further processing.

On the router named cisco-on-tymnet, you enter these commands:

```
x25 route vax-x121-address ip cisco-on-vax
x25 route .* interface serial 0
```

These commands force all calls with the VAX X.121 address to be sent to the router with the VAX connected to it. All other calls with X.121 addresses are forwarded out to Tymnet. If Tymnet can route them, a Call Accepted packet is returned, and everything proceeds normally. If Tymnet cannot handle the calls, it clears each call and the Clear Request packet is forwarded back toward the VAX.

Configure XOT (Remote) X.25 Route

A remote X.25 route is one that crosses a TCP connection. Such routes are called X.25-over-TCP, or XOT, routes. (XOT was previously called remote routes or tunneled routes.

When an incoming call needs to be forwarded, two fields in the X.25 routing table are consulted to determine a remote X.25 route:

- Destination X.121 address
- X.25 packet's Call User Data (CUD) field (optional)

When the destination address and the CUD of the incoming packet fit the X.121 and CUD patterns in the routing table, the call is forwarded.

You can also specify an XOT source that causes the XOT TCP con nection to use the IP address of a specified interface as the source address of the TCP connection. If, for instance, a loopback interface is specified for the XOT connection's source address, TCP can use a primary interface or any backup interface to reach the other end of th connection. However, if a physical interface address is specified as the source address, the XOT connection is terminated if that interface goes down.

In global configuration mode, configure an XOT route (thus adding it to the X.25 routing table):

x25 route [*#position*] *x121-address* [cud pattern] ip *ip-address* [xot-source *type number*]

Configure a Locally Switched PVC

You can configure an X.25 PVC in the X.25 switching software. As a result, DTEs that require permanent circuits can be connected to router acting as an X.25 switch and have a properly functioning connection. X.25 resets will be sent to indicate when the circuit comes up or goes down. Both interfaces must define complementary locally switched PVCs.

To configure a locally switched PVC, type (in interface configuration mode):

x25 pvc *number1* interface *type number* pvc *number2* [*option*]

The command options ar **packetsize *in out*** and **windowsize *in out***; they allow a PVC's flow control values to be defined if they differ from the interface defaults.

Example: PVC Switching on the Same Router

In this example, a PVC is connected between two serial interfaces on the same router. In this type of interconnection configuration, the destination interface must be specified along with the PVC number on that interface. To make a working PVC connection, two commands must be specified, each pointing to the other.

```
interface serial 0
encapsulation x25
x25 ltc 5
x25 pvc 1 interface serial 1 pvc 4
!
interface serial 1
encapsulation x25
x25 ltc 5
x25 pvc 4 interface serial 0 pvc
```

In global configuration mode, ensure that these TCP sessions remain connected in the absence of XOT traffic by enabling keepalives:

service tcp-keepalives-in

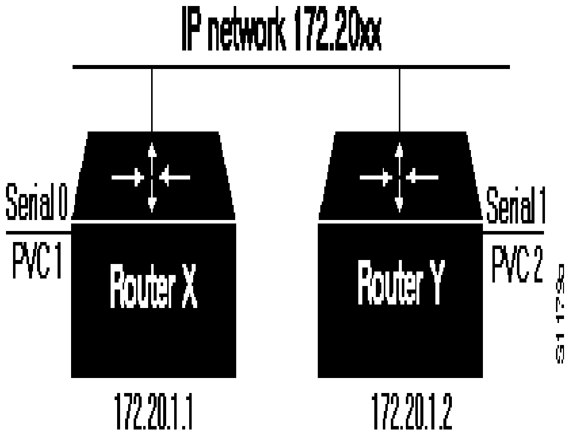
service tcp-keepalives-out

TCP keepalives also inform a router when an XOT SVCs session is not active, thus freeing router resources.

Example: Simple Remote PVC Tunneling

In this example, a connection is established between two PVCs across a LAN. Because the connection is remote (across the LAN), the tunneling command is used. This example establishes a PVC between Router X, Serial 0, PVC 1 and Router Y, Serial 1, PVC 2. Keepalives are enabled to maintain connection notification. Figure 12-4 provides a visual representation of the configuration.

Figure 12-4:X.25 Tunneling Connection



Configuration for Router X

```
service tcp-keepalives-in
service tcp-keepalives-out
interface serial 0
x25 pvc 1 tunnel 172.20.1.2 interface serial 1 pvc 2
```

Configuration for Router Y

```
service tcp-keepalives-in
service tcp-keepalives-out
interface serial 1
x25 pvc 2 tunnel 172.20.1.1 interface serial 0 pvc 1
```

Configure an XOT (Remote) PVC

A PVC can be connected to another router over a LAN with the XOT protocol. When the interfaces come up, a TCP connection is established to the router that is acting as the switch for the destination. All X.25 packets will be sent and received over this reliable data stream.

Flow control is maintained end-to-end. This was previously called remote switching or tunneling.

Running X.25 over TCP/IP provides a number of benefits. Other routers can switch IP datagrams containing the X.25 packets using the router's high-speed switching abilities. X.25 data can be sent over networks running only TCP/IP protocols. The TCP/IP protocol suite runs over many different networking technologies, including Ethernet, Token Ring, T1 serial, and FDDI. Thus X.25 data can be forwarded over these media to another XOT host where it can be output to an X.25 interface. Both interfaces must define complementary tunneled PVCs.

To configure a remote PVC to connect across a TCP/IP LAN, type (in interface configuration mode):

```
x25 pvc number1 tunnel address interface
serial string pvc number2 [option]
```

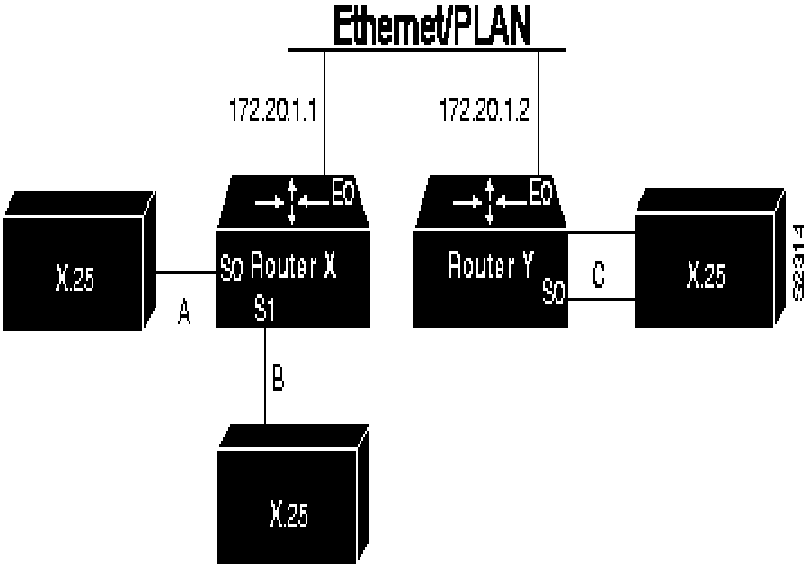
The command options ar **`packetsize`** ***`in out`*** and **`window-size`** ***`in out`***; they allow a PVC's flow control values to be defined if they differ from the interface defaults.

Each XOT connection relies on a TCP session to carry traffic. If you do not enable TCP keepalives, XOT PVCs might encounter problems if one end of the connection is reloaded. When the reloaded host attempts to establish a new connection, the other host refuses the new connection because it has not been informed that the old session is no longer active. Recovery from this state requires the other host to be informed that its TCP session is no longer viable so that it attempts to reconnect the PVC.

Example: Remote PVC Tunneling

In the more complex example shown in Figure 12-5, the connection between points A and B is switched, and the connections between point C and points A and B are tunneled. Keepalives are enabled to maintain connection notification.

Figure 12-5:Local Switching and Remote Tunneling PVCs



Configuration for Router X

```
service tcp-keepalives-in
service tcp-keepalives-out
interface ethernet 0
ip address 172.20.1.1 255.255.255.0
!
interface serial 0
x25 ltc 5
x25 pvc 1 interface serial 1 pvc 1
x25 pvc 2 tunnel 172.20.1.2 interface serial 0 pvc 1
!
interface serial 1
x25 ltc 5
x25 pvc 1 interface serial 0 pvc 1
x25 pvc 2 tunnel 172.20.1.2 interface serial 0 pvc 2
```

Configuration for Router Y

```
service tcp-keepalives-in
service tcp-keepalives-out
interface ethernet 0
ip address 172.20.1.2 255.255.255.0
!
interface serial 0
x25 ltc 5
x25 pvc 1 tunnel 172.20.1.1 interface serial 0 pvc 2
x25 pvc 2 tunnel 172.20.1.1 interface serial 1 pvc 2
```

Configure Additional X.25 Routing Features

To configure other, less common X.25 routing features, perform these tasks:

- Configure XOT to Use Interface Default Flow Control Values
- Substitute Addresses in a Local X.25 Rout
- Configure XOT Alternate Destinations

Configure XOT to Use Interface Default Flow Control Values

When setting up a connection, the source and destination XOT implementations need to cooperate to determine the flow control values that apply to the SVC. The source XOT ensures cooperation by encoding the X.25 flow control facilities (the window sizes and maximum packet sizes) in the X.25 Call packet. The far host's XOT implementation can then correctly negotiate the flow control values at the destination interface and, if needed, indicate the final values in the X.25 Call Confirm packet.

The versions of XOT prior to Release 9.1(4.1) software will not, however, ensure that these flow control values are encoded in the X.25 Call packet. When XOT receives a call that leaves one or both flow control values unspecified, it supplies the values. The values supplied are a window size of 2 packets and maximum packet size of 128 bytes; ac cording to the standards, any SVC can be negotiated to use these values. Thus, when XOT receives a call from an older XOT implementation, it can specify in the Call Confirm packet that thes flow control values must revert to the lowest common denominator.

What the older XOT implementations required was that the source and destination XOT router use the same default flow control values on the two X.25 interfaces that connect the SVC. Consequently, connections with mismatched flow control values were created when this assumption was not true, resulting in mysterious problems. The current implementation's practice of signaling the values in the Call Confirm packet avoids these problems.

Occasionally the older XOT implementation will be connected to a piece of X.25 equipment that cannot handle modification of the flow control parameters in the Call Confirm packet. These configurations should be upgraded to use a more recent version of XOT; when up grade is not possible, XOT's behavior causes a migration problem. In this situation, you may configure the Cisco IOS software to cause XOT to obtain unspecified flow control facility values from the destination interface's default values.

Modify XOT's source of unencoded flow control values by adding the option **`use-tcp-if-defs`** when enabling X.25 routing in global configuration mode:

```
x25 routing [use-tcp-if-defs]
```

Substitute Addresses in a Local X.25 Route

When interconnecting two separate X.25 networks, you must sometimes provide for address translation for local routes. Your X.25 switch supports translation of X.25 source and destination addresses for local switching.

To translate addresses, perform one or both of these tasks in global configuration mode:

- Translate the X.25 source address for local switching.

  ```
  x25 route [#position] x121-address
  [substitute-source pattern] [cud pattern]
  interface interface number
  ```

- Translate the X.25 destination address for local switching.

  ```
  x25 route [#position] x121-address
  [substitute-dest pattern] [cud pattern]
  interface interface number
  ```

Address substitution is not available for XOT routes.

Configure XOT Alternate Destinations

XOT routes can be configured with alternate addresses. On routing call, XOT will try each XOT destination host in sequence; if the TCP connection establishment fails, the next destination will be tried. Up to six XOT destination addresses can be entered.

To configure an XOT route with alternate addresses, thus adding it to the X.25 routing table, type (in global configuration mode):

```
x25 route [#position] x121-address [cud
pattern]

ip ip-address [ip-address2... [ip-
address6]]
```

The sequence of alternate destination XOT host addresses is simply added to the normal XOT route configuration command.

Note: ***It can take up to 50 seconds to try an alternate route due to TCP timings.***

Configure CMNS Routing

The Connection-Mode Network Service (CMNS) provides a mecha nism through which local X.25 switching can be extended to nonserial media through the use of OSI-based NSAP addresses. This implementation runs packet-level X.25 over frame-level LLC2.

The AI2524 CMNS implementation allows LAN-based OSI re sources, such as a DTE host and a Sun workstation, to be interconnected to each other via the router's LAN interfaces and to a remot OSI-based DTE through a WAN interface using, for example, an X.25 Packet-Switched Network (PSN).

Note: ***CMNS is implicitly enabled whenever an X.25 encapsulation is included with a serial interfac configuration.***

All local mapping is performed by the static mapping of MAC addresses and X.121 addresses to NSAP addresses.

Implementing CMNS routing involves completing these tasks:

- Enable CMNS on an Interface
- Specify a CMNS Static Map of Addresses

Enable CMNS on an Interface

In interface configuration mode, enable CMNS on a nonserial interface:

```
cmns enable
```

Specify a CMNS Static Map of Addresses

After enabling CMNS on a nonserial interface (or specifying X.25 encapsulation on a serial interface), you must map NSAP addresses to either MAC-layer addresses or X.121 addresses, depending on th application.

For CMNS support over dedicated serial links (such as leased lines), an X.121 address is not needed, but can be included. You must specify the X.121 address for CMNS connections over a packet-switched network, and you must specify a MAC address for CMNS connections over a nonserial medium (Ethernet, FDDI, or Token Ring).

To map the NSAP addresses to either a MAC address or X.121 ad dress, perform one of these tasks in interface configuration mode:

- Statically map an NSAP address to a nonserial MAC-layer address.

  `x25 map cmns` ***`nsap mac-address`***

- Statically map an NSAP address to X.25, with an optional X.121 destination address.

  `x25 map cmns` ***`nsap x121-address`***

Example: CMNS Configured for X.121 and MAC Addresses

This example illustrates enabling CMNS and configuring X.121 and MAC address mappings. Map NSAP to MAC-address on Ethernet0:

```
interface ethernet 0
cmns enable
x25 map cmns 38.8261.1000.0150.1000.17 0000.0c00.ff89
```

Map NSAP to X.121-address on Serial0 assuming the link is over PDN:

```
interface serial 0
encapsulation x25
x25 map cmns 38.8261.1000.0150.1000.18 3110451
```

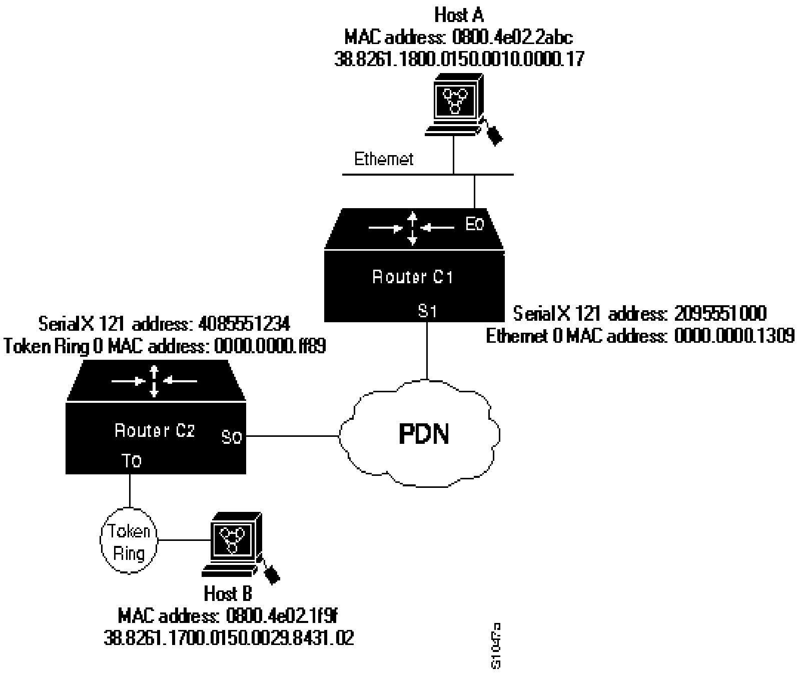
Specify cmns support for Serial1 assuming that the link is over leased line:

```
interface serial 1
encapsulation x25
x25 map cmns 38.8261.1000.0150.1000.20
```

Example: CMNS Switched over a PDN

This example depicts switching CMNS over a packet-switched PDN. Figure 12-6 illustrates the general network topology for a CMNS switching application where calls are being made between resources on opposite sides of a remote link to Host A (on an Ethernet) and Host B (on a Token Ring), with a PDN providing the connection.

Figure 12-6:Example Network Topology for Switching CMNS over a PDN


Host A
MAC address: 0800.4e02.2abc
38.8261.1800.0150.0010.0000.17
Ethernet
E0
Router C1
S1
SerialX 121 address: 2095551000
Ethernet 0 MAC address: 0000.0000.1309
SerialX 121 address: 4085551234
Token Ring 0 MAC address: 0000.0000.ff89
Router C2
SO
To
PDN
Token Ring
Host B
MAC address: 0800.4e02.1f9f
38.8261.1700.0150.0029.8431.02
S1047a

This configuration listing allows resources on either side of the PD to call Host A or Host B. This configuration allows traffic intended for the remote NSAP address specified in the **`x25 map cmns`** commands (for the serial ports) to be switched through the serial interface for which CMNS is configured.

Configuration for Router C2

This configuration specifies that any traffic from any other interface intended for any NSAP address with NSAP prefix 38.8261.17 will be switched to MAC address 0800.4e02.1f9f through Token Ring 0.

```
interface token 0
cmns enable
x25 map cmns 38.8261.17 0800.4e02.1f9f
```

This configuration specifies that traffic from any other interface on Cisco Router C2 that is intended for any NSAP address with NSAP-prefix 38.8261.18 will be switched to X.121 address 2095551000 through Serial 0.

```
interface serial 0
encapsulation x25
x25 address 4085551234
x25 map cmns 38.8261.18 2095551000
```

Configuration for Router C1

This configuration specifies that any traffic from any other interface intended for any NSAP address with NSAP 38.8261.18 will be switched to MAC address 0800.4e02.2abc through Ethernet 0.

```
interface ethernet 0
cmns enable
x25 map cmns 38.8261.18 0800.4e02.2abc
```

This configuration specifies that traffic from any other interface on Cisco Router C1 that is intended for any NSAP address with NSAP-prefix 38.8261.17 will be switched to X.121 address 4085551234 through Serial 1.

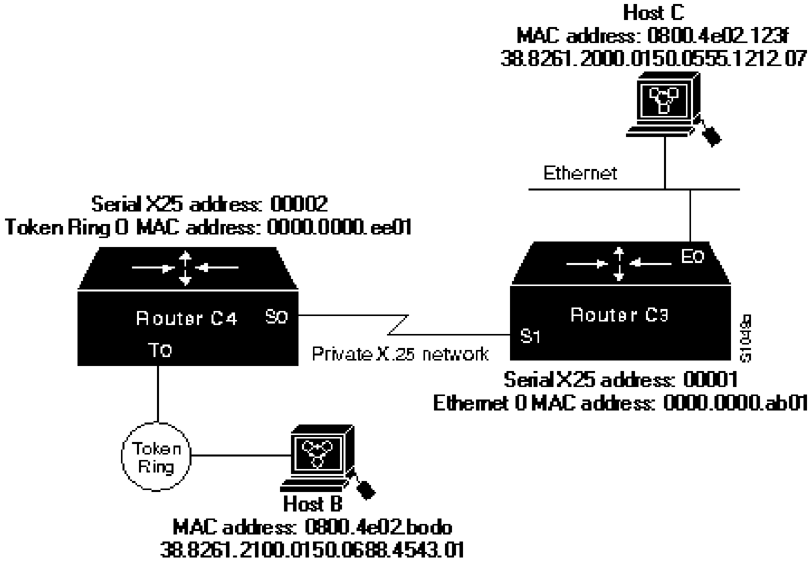
```
interface serial 1
encapsulation x25
x25 address 2095551000
x25 map cmns 38.8261.17 4085551234
```

Example: CMNS Switched over Leased Lines

This example illustrates switching CMNS over a leased line. Figur 12-7, illustrates the general network topology for a CMNS switching application where calls are being made by resources on the opposit sides of a remote link to Host C (on an Ethernet) and Host B (on a Token Ring), with a dedicated leased line providing the connection.

This configuration listing allows resources on either side of the leased line to call Host C or Host B. This configuration allows traffic intended for the remote NSAP address specified in the **`x25 map cmns`** commands (for the serial ports) to be switched through the serial interface for which CMNS is configured.

Figure 12-7:Example Network Topology for Switching CMNS over a Leased Line


Host C
MAC address: 0800.4e02.123f
38.8261.2000.0150.0555.1212.07
Ethernet
Serial X25 address: 00002
Token Ring 0 MAC address: 0000.0000.ee01
E0
Router C4
S0
T0
Router C3
S1
Private X.25 network
S1049
Serial X25 address: 00001
Ethernet 0 MAC address: 0000.0000.ab01
Token Ring
Host B
MAC address: 0800.4e02.bodo
38.8261.2100.0150.0688.4543.01

A key difference for this configuration compared with the previous example is that with no PDN, the specification of an X.121 address in the **`x25 map cmns`** command is not necessary. The specification of an X.25 address also is not needed, but is included for symmetry with the previous example.

Configuration for Router C4

This configuration specifies that any traffic from any other interface intended for any NSAP address with NSAP 38.8261.21 will be switched to MAC address 0800.4e02.bcd0 through Token Ring 0.

```
interface token 0
cmns enable
x25 map cmns 38.8261.21 0800.4e02.bcd0
```

This configuration specifies that traffic from any other interface on Cisco Router C4 that is intended for any NSAP address with NSAP-prefix 38.8261.20 will be switched through Serial 0.

```
interface serial 0
encapsulation x25
x25 address 00002
x25 map cmns 38.8261.20
```

Configuration for Router C3

This configuration specifies that any traffic from any other interface intended for any NSAP address with NSAP 38.8261.20 will be switched to MAC address 0800.4e02.123f through Ethernet 0.

```
interface ethernet 0
cmns enable
x25 map cmns 38.8261.20 0800.4e02.123f
```

This configuration specifies that traffic from any other interface on Router C3 that is intended for any NSAP address with NSAP-prefix 38.8261.21 will be switched through Serial 1.

```
interface serial 1
encapsulation x25
x25 address 00001
x25 map cmns 38.8261.21
```

Create X.29 Access Lists

Protocol translation software supports access lists, which make it possible to limit access to the access server from X.25 hosts. Access lists take advantage of the message field defined by Recommendation X.29, which describes procedures for exchanging data between two PADs or between a PAD and a DTE device.

To define X.29 access lists, perform these tasks:

- Create an Access List
- Apply an Access List to a Line

When configuring protocol translation, you can specify an access list number with each **translate** command. When translation sessions result from incoming PAD connections, the corresponding X.29 ac cess list is used.

Create an Access List

To specify the access conditions, restrict incoming and outgoing connections between a particular Virtual Terminal (VTY) line (into a Cisco access server) and the addresses in an access list by typing (in global configuration mode):

```
x29 access-list access-list-number {deny |
permit} x121-address
```

An access list can contain any number of lines. The lists are processed in the order in which you type the entries. The first match causes the permit or deny condition. If an X.121 address does not match any of the entries in the access list, access is denied.

Example: X.29 Access List

This example illustrates an X.29 access list. Incoming permit conditions are set for all IP hosts that have specific characters in their names. All X.25 connections to a printer are denied. Outgoing connections are list restricted.

Permit all IP hosts beginning with VMS. Deny X.25 connections to th printer on line 5.

```
access-list 1 permit 0.0.0.0 255.255.255.255
lat access-list 1 permit ^VMS.*
x29 access-list 1 deny .*
!
line vty 5
 access-class 1 i
```

Permit outgoing connections for other lines. Permit IP access with the network 172.30.

```
 access-list 2 permit 172.30.0.0 0.0.255.255
```

Permit X.25 connections to Infonet hosts only.

```
 x29 access-list 2 permit ^31370
!
line vty 0 16
 access-class 2 ou
```

Apply an Access List to a Line

In line configuration mode, apply an access list to a virtual line by restricting incoming and outgoing connections between a particular virtual terminal line (into a Cisco access server) and the addresses in an access list:

```
access-class access-list-number in
```

The access list number is used for incoming TCP access and for incoming PAD access. For TCP access, the protocol translator uses the defined IP access lists. For incoming PAD connections, the protocol translator uses the defined X.29 access list. If you want to have access restrictions only on one of the protocols, create an access list that permits all addresses for the other protocols.

Create an X.29 Profile Script

You can create an X.29 profile script for th **translate** command. When an X.25 connection is established, the protocol translator acts as if an X.29 Set Parameter packet had been sent containing the parameters and values set by this command.

In global configuration mode, create an X.29 profile script:

```
x29 profile name parameter:value
[parameter:value]
```

Example: X.29 Profile Script

This profile script turns local edit mode on when the connection is made and establishes local echo and line termination upon receipt of a Return. The name linemode is used with the **translate** command to activate this script.

```
x29 profile linemode 2:1 3:2 15:1
translate tcp 172.30.1.26 x25 55551234 profile linemode
```

Configure LAPB

You can use only LAPB as a serial encapsulation method if you hav a private serial line. You must use one of the X.25 packet-level encapsulations when attaching to an X.25 network.

The LAPB standards distinguish between two types of hosts: Data Terminal Equipment (DTE), and Data Circuit-terminating Equipment (DCE). At Level 2, or the data link layer in the OSI model, LAPB allows for orderly and reliable exchange of data between a DTE and a DCE. A router using LAPB encapsulation can act as a DTE or DCE device at the protocol level, which is distinct from the hardware DTE or DCE identity.

Using LAPB under noisy conditions can result in greater throughput than HDLC encapsulation. When LAPB detects a missing frame, th router retransmits the frame instead of waiting for the higher layers to recover the lost information. This behavior is good only if the host timers are relatively slow. In the case of quickly expiring host timers, however, you will discover that LAPB is spending much of its tim transmitting host retransmissions. If the line is not noisy, the lower overhead of HDLC encapsulation is more efficient than LAPB. When you are using long-delay satellite links, for example, the lock-step behavior of LAPB makes HDLC encapsulation the better choice.

To configure LAPB, complete these tasks. The tasks in the first section are required; the remaining are optional.

- Configure a LAPB Datagram Transport
- Modify LAPB Protocol Parameters
- Configure LAPB Priority and Custom Queuing
- Configure Transparent Bridging over Multiprotocol LAPB
- Monitor and Maintain LAPB and X.25

Configure a LAPB Datagram Transport

Set the appropriate LAPB encapsulation to run datagrams over a serial interface. One end of the link must be DTE, the other must be DCE.

1. Specify a serial interface.

   `interface serial` ***`number`***

2. In interface configuration mode, select an encapsulation and protocol if using a single protocol or select the multiple protocol operation by performing one or more of these tasks:

   - Enable encapsulation of a single protocol on the line using DCE operation.

   encapsulation lapb dce [*protocol*]

   - Enable encapsulation of a single protocol on the line using DTE operation.

   encapsulation lapb [dte] [*protocol*]

   - Enable use of multiple protocols on the line using DCE opera tion.

   encapsulation lapb dce multi

   - Enable use of multiple protocols on the line using DTE operation.

   encapsulation lapb [dte] multi

Single protocol LAPB defaults to IP encapsulation. Multiprotocol LAPB does not support source-route bridging or TCP/IP header compression, but does support transparent bridging. Only protocols supported by a single protocol encapsulation are supported by multiprotocol LAPB encapsulation.

Example: Typical LAPB Configuration

In this example, the frame size (N1), window size (k), and maximum retransmission (N2) parameters retain their default values. The **encapsulation** interface configuration command sets DCE operation to carry a single protocol, IP by default. The **lapb t1** interfac configuration command sets the retransmission timer to 4,000 milliseconds (4 seconds) for a link with a long delay or slow connecting DTE device.

```
interface serial 3
encapsulation lapb dce
lapb t1 4000
```

Modify LAPB Protocol Parameters

X.25 Level 2 or LAPB operates at the data link layer of the OSI reference model. LAPB specifies methods for exchanging data (in units called frames), detecting out-of-sequence or missing frames, retrans mitting frames, and acknowledging frames. Several protocol parameters can be modified to change LAPB protocol performance on particular link. Because X.25 operates the PLP on top of the LAPB protocol, these tasks apply to both X.25 links and LAPB links. The parameters and their default values are summarized in this table.

Task (LAPB Parameter)	Command	Values or Ranges	Default
Set the modulo.	**lapb modulo** ***modulus***	8 or 128	8
Set the window size (K).	**lapb k** ***window-size***e	1- (modulo minus 1) frames	7
Set the maximum bits per frame (N1).	**lapb n1** ***bits***	Bits (must be a multiple of 8)	Based on hardware MTU and protocol overhead
Set the count for sending frames (N2).	**lapb n2** ***tries***	1--255 tries	20
Set the retransmission timer (T1).	**lapb t1** ***milliseconds***	1--64000 milliseconds	3000
Set the hard ware outage period.	**lapb interface-outage** ***milliseconds***		0 (disabled)
Set the idle link period (T4).	**lapb t4** ***seconds***		0 (disabled)

- LAPB Modulo and LAPB K. The LAPB modulo determines the operating mode. Modulo 8 (basic mode) is widely available, because it is required for all standard LAPB implementations and is sufficient for most links. Modulo 128 (extended mode) can achieve greater throughput on high-speed links that have a low error rate (some satellite links, for example) by increasing the number of frames that can be transmitted before waiting for acknowledgment (as configured by the LAPB window parameter, k). By its design, LAPB's k parameter can be at most one less than the operating modulo. Modulo 8 links can typically send seven frames before an acknowledgment must be received; modulo 128 links can set k to a value as large as 127. By default, LAPB links use the basic mode with a window of 7.
- LAPB N1. When connecting to an X.25 network, use the N1 parameter value set by the network administrator. This value is the maximum number of bits in an LAPB frame, which determines the maximum size of an X.25 packet. When you are using LAPB over leased lines, the N1 parameter should be eight times the hardwar Maximum Transmission Unit (MTU) size plus any protocol overhead.

  The LAPB N1 range is dynamically calculated by the Cisco IOS software whenever an MTU change, an L2/L3 modulo change, o a compression change occurs on a LAPB interface.

Caution: ***The LAPB N1 parameter provides little benefit beyond the interface MTU and can easily cause link failures if misconfigured. This parameter should be left at its default value.***

- LAPB N2. The transmit counter (N2) is the number of unsuccessful transmit attempts made before the link is declared down.
- LABP T1. The retransmission timer (T1) determines how long transmitted frame can remain unacknowledged before the Cisco IOS software polls for an acknowledgment. For X.25 networks, the retransmission timer setting should match that of the network.

  For leased-line circuits, the T1 timer setting is critical because the design of LAPB assumes that a frame has been lost if it is not acknowledged within period T1. The timer setting must be larg enough to permit a maximum-sized frame to complete one round trip on the link. If the timer setting is too small, the software will poll before the acknowledgment frame can return, which may result in duplicated frames and severe protocol problems. If the timer setting is too large, the software waits longer than necessary before requesting an acknowledgment, which reduces bandwidth.

- LAPB T4. The LAPB standards define a timer to detect unsignaled link failures (T4). The T4 timer is reset every time a frame is received from the partner on the link. If the T4 timer expires, Receiver Ready frame with the Poll bit set is sent to the partner, which is required to respond. If the partner does not respond, the standard polling mechanism is used to determine whether the link is down. The period of T4 must be greater than the period of T1.

  Another LAPB timer function allows brief hardware failures while the protocol is up, without requiring a protocol reset. When a brief hardware outage occurs, the link will continue uninterrupted if the outage is corrected before the specified hardware outage period expires.

Configure LAPB Priority and Custom Queuing

AI2524 priority queuing and custom queuing are available for LAPB to allow you to improve link responsiveness to a given type of traffic by specifying the priority of that type of traffic for transmission on the link.

Priority queuing is a mechanism that classifies packets based on certain criteria and then assigns the packets to 1of 4 output queues, with high, medium, normal, or low priority. Custom queuing similarly classifies packets, assigns them to 1 of 10 output queues, and controls the percentage of an interface's available bandwidth that is used for a queue.

For example, you can use priority queuing to ensure that all Telnet traffic is processed promptly and that Simple Mail Transfer Protocol (SMTP) traffic is sent only when there is no other traffic to send. Pri ority queuing in this example can starve the non-Telnet traffic; custom queuing can be used instead to ensure that some traffic of all categories is sent.

Both priority queuing and custom queuing can be defined, but only one method can be assigned to a given interface.

To configure priority and custom queuing for LAPB, perform thes tasks:

1. Perform the standard priority and custom queuing tasks except the task of assigning a priority or custom group to the interface.

2. Perform the standard LAPB encapsulation tasks, as specified in the Configure a LAPB Datagram Transport section of this chapter.

3. Assign either a priority group or a custom queue to the interface.

Note: ***The*** **`lapb hold-queue`** ***command is no longer supported, but the same functionality is provided by the*** **`hold-queue`** ***`size`*** **`out`** ***standard queue control command.***

Configure Transparent Bridging over Multiprotocol LAPB

To configure transparent bridging over multiprotocol LAPB, perform these tasks beginning in global configuration mode:

1. Specify the serial interface, and enter interface configuration mode.

   `interface serial` *`number`*

2. Assign no IP address to the interface.

   `no ip address`

3. Configure multiprotocol LAPB encapsulation.

   `encapsulation lapb multi`

4. Assign the interface to a bridge group.

   `bridge-group` *`bridge-group`*

5. Define the type of spanning tree protocol.

   `bridge bridge-group protocol {ieee|dec}`

Note: ***This feature requires use of the*** **`encapsulation lapb multi`** ***command. You cannot use the*** **`encapsulation lapb protocol`** ***command with a*** **`bridge`** ***keyword to configure this feature.***

Monitor and Maintain LAPB and X.25

To monitor and maintain X.25 and LAPB, perform any of these tasks in EXEC mode:

- Clear all virtual circuits at once (everything—encapsulation, routed calls, and PAD calls—is cleared), or clear the single virtual circuit specified.

  `clear x25-vc` *`type number`* **`[`***`lcn`***`]`**

- Display CMNS information.

  `show cmns [`*`type number`***`]`**

- Display operation statistics for an interface.

  `show interfaces serial` *`number`*

- Display CMNS connections over LLC2.

  `show llc2`

- Display the protocol-to-X.121 address map.

  `show x25 map`

- Display routes assigned by the **`x25 route`** command.

  `show x25 route`

- Display details of active virtual circuits.

  `show x25 vc [lcn]`

Example: Transparent Bridging for Multiprotocol LAPB Encapsulation

This example configures transparent bridging for multiprotocol LAPB encapsulation:

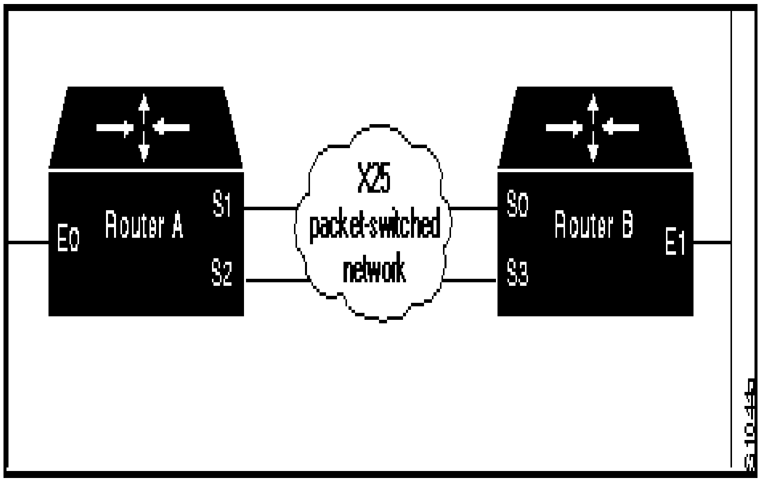
```
no ip routing
!
interface Ethernet 1
no ip address
no mop enabled
bridge-group 1
!
interface serial 0
no ip address
encapsulation lapb multi
bridge-group 1
!
bridge 1 protocol ieee
```

Example: X.25 Configured to Allow Ping Support over Multiple Lines

For **`ping`** commands to work in an X.25 environment (when load sharing over multiple serial lines), you must include entries for all adjacent interface IP addresses in the **`x25 map`** command for each serial interface. This example illustrates this point.

Consider two routers, Router A and Router B, communicating with each other over two serial lines via an X.25 PDN (see Figure 12-8) or over leased lines. In either case, all serial lines must be configured fo the same IP subnet address space. The configuration that follows allows for successful **`ping`** commands. A similar configuration is re quired for the same subnet IP addresses to work across X.25.

Figure 12-8:Parallel Serial Lines to an X.25 Network



Note: ***All four serial ports configured for the two routers in the following configuration example must be assigned to the same IP subnet address space. In this case, the subnet is 172.20.170.0.***

Configuration for Router A

```
interface serial 1
ip 172.20.170.1 255.255.255.0
x25 address 31370054068
x25 map ip 172.20.170.3 31370054065
x25 map ip 172.20.170.4 31370054065
!
interface serial 2
ip 172.20.170.2 255.255.255.0
x25 address 31370054069
x25 map ip 172.20.170.4 31370054067
x25 map ip 171.20.170.3 31370054067
```

(allow either destination address)

```
x25 31370054068 alias serial2
x25 31370054069 alias serial1
```

Configuration for Router B

```
interface serial 0
ip 172.20.170.3 255.255.255.0
x25 address 31370054065
x25 map ip 172.20.170.1 31370054068
x25 map ip 172.20.170.2 31370054068
!
interface serial 3
ip 172.20.170.4 255.255.255.0
x25 address 31370054067
x25 map ip 172.20.170.2 31370054069
x25 map ip 172.20.170.1 31370054069
```

(allow either destination address)

```
x25 31370054065 alias serial3
x25 31370054067 alias serial0
```

Example: Booting from a Network Server over X.25

Over X.25, you cannot boot the router from a network server via a broadcast. Instead, you must boot from a specific host. Also, an **`x25 map`** command must exist for the host that you boot from. The **`x25 map`** command is used to map an IP address into an X.121 address. There must be an **`x25 map`** command that matches the IP address given on the **`boot system`** command line.

```
boot system gs3-k.100 172.18.126.111
interface Serial 1
ip address 172.18.126.200 255.255.255.0
encapsulation X25
x25 address 10004
x25 map IP 172.18.126.111 10002 broadcast
lapb n1 12040
clockrate 56000
```

In this case, 10002 is the X.121 address of the remote router that can get to host 172.18.126.111.

The remote router must have this **`x25 map`** entry:

```
x25 map IP 172.18.126.200 10004 broadcast
```

This entry allows the remote router to return a boot image from the host to the router booting over X.25.

Chapter 13: AI2524 Frame Relay Configuration Steps

Introduction

This chapter describes how to perform a variety of frame relay configuration tasks and enable frame relay encapsulation.

Frame Relay Hardware Configuration

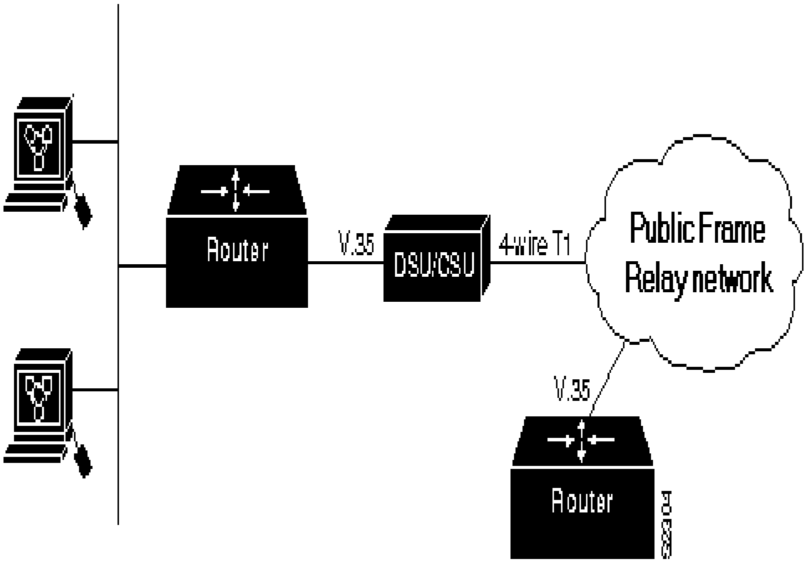
Routers and access servers can connect directly to:

- The Frame Relay switch
- A channel service unit/digital service unit (CSU/DSU), which then connects to a remote Frame Relay switch

A Frame Relay network is not required to support only routers that are connected directly or only routers connected via CSU/DSUs. Within network, some routers can connect to a Frame Relay switch through direct connection and others through connections via CSU/DSUs. However, a single router interface configured for Frame Relay can be only one or the other.

The CSU/DSU converts V.35 or RS-449 signals to the properly coded T1 transmission signal for successful reception by the Frame Relay network. Figure 13-1 illustrates the connections between the different components.

Figure 13-1:Typical Frame Relay Configuration



The Frame Relay interface actually consists of one physical connection between the network server and the switch that provides the service. This single physical connection provides direct connectivity to each device on a network, such as a StrataCom FastPacket wide-area network (WAN).

Frame Relay Configuration Task List

There are required, basic steps you must follow to enable Frame Relay for your network. In addition, you can customize Frame Relay for you particular network needs and monitor Frame Relay connections. These sections outline these tasks. The tasks in the first two sections are required.

- Enable Frame Relay Encapsulation on an Interface
- Configure Dynamic or Static Address Mapping
- Configure the LMI
- Configure Frame Relay Switched Virtual Circuits
- Configure Frame Relay Traffic Shaping
- Customize Frame Relay for Your Network
- Monitor the Frame Relay Connections

Enable Frame Relay Encapsulation on an Interface

Beginning in global configuration mode, set Frame Relay encapsulation at the interface level:

1. Specify the serial interface, and enter interface configuration mode.

   ```
   interface serial number
   ```

2. Enable Frame Relay, and specify the encapsulation method.

   ```
   encapsulation frame-relay [ietf]
   ```

Frame Relay supports encapsulation of all supported protocols in conformance with RFC 1490, allowing interoperability between multiple vendors. Use the Internet Engineering Task Force (IETF) form of Frame Relay encapsulation if your router or access server is connected to another vendor's equipment across a Frame Relay network. IETF encapsulation is supported either at the interface level or on a per-virtual circuit basis.

Examples: IETF Encapsulation

The first example sets IETF encapsulation at the interface level. The second example sets IETF encapsulation on a per-DLCI basis. In the first example, the keyword ietf sets the default encapsulation method for all maps to IETF.

```
encapsulation frame-relay IETF
frame-relay map ip 131.108.123.2 48 broadcast
frame-relay map ip 131.108.123.3 49 broadcast
```

In this example, IETF encapsulation is configured on a per-DLCI basis. This configuration has the same result as the configuration in th first example.

```
encapsulation frame-relay
frame-relay map ip 131.108.123.2 48 broadcast ietf
frame-relay map ip 131.108.123.3 49 broadcast ietf
```

Configure Dynamic or Static Address Mapping

Dynamic address mapping uses Frame Relay Inverse ARP to request the next hop protocol address for a specific connection, given its known DLCI. Inverse ARP is enabled by default for all protocols it supports, but can be disabled for specific protocol-DLCI pairs. As a result, you can use dynamic mapping for some protocols and static mapping for other protocols on the same DLCI.

Configure Dynamic Mapping

Inverse ARP is enabled by default for all protocols enabled on the physical interface. Packets are not sent out for protocols that are not enabled on the interface.

Because Inverse ARP is enabled by default, no additional command is required to configure dynamic mapping on an interface.

Configure Static Mapping

A static map links a specified next hop protocol address to a specified DLCI. Static mapping removes the need for Inverse ARP requests; when you supply a static map, Inverse ARP is automatically disabled for the specified protocol on the specified DLCI.

To establish static mapping according to your network needs, perform one of these tasks in interface configuration mode:

- Define the mapping between a next hop protocol address and the DLCI used to connect to the address.

  `frame-relay map` ***`protocol protocol-address`*** **`dlci [broadcast] [ietf] [cisco]`**

- Define a DLCI used to send International Organization for Standardization (ISO) Connectionless Network Service (CLNS) frames.

  `frame-relay map clns dlci [broadcast]`

- Define a DLCI used to connect to a bridge.

  `frame-relay map bridge dlci [broadcast] [ietf]`

Use these keywords to specify the protocols:

- **`ip`**—IP
- **`ipx`**—Novell IPX
- **`clns`**—ISO CLNS

Examples: Static Address Mapping

These sections provide examples of static address mapping for the IP and IPX protocols.

Two Routers in Static Mode Example

This example illustrates how to configure two routers for static mode.

Configuration for Router 1

```
interface serial 0
ip address 131.108.64.2 255.255.255.0
encapsulation frame-relay
keepalive 10
frame-relay map ip 131.108.64.1 43
```

Configuration for Router 2

```
interface serial 0
ip address 131.108.64.1 255.255.255.0
encapsulation frame-relay
keepalive 10
frame-relay map ip 131.108.64.2 43
```

IPX Routing Example

This example illustrates how to send packets destined for IPX address 200.0000.0c00.7b21 out on DLCI 102:

```
ipx routing 000.0c00.7b3b
!
interface ethernet 0
ipx network 2abc
!
interface serial 0
ipx network 200
encapsulation frame-relay
frame-relay map ipx 200.0000.0c00.7b21 102 broadcast
```

Configure the LMI

Local Management Interface (LMI) autosense enables the interface to determine the LMI type supported by the switch and eliminates the need to configure the Local Management Interface (LMI) explicitly.

Allow LMI Autosense to Operate

LMI autosense is active in these situations:

- The router is powered up or the interface changes state to up.
- The line protocol is down but the line is up.
- The interface is a Frame Relay DTE.
- The LMI type is not explicitly configured.

The LMI Autosense Process

When LMI autosense is active, it sends out a full status request, in all 3 LMI options (ANSI, ITU, Cisco) to the switch.

The router decodes the reply and configures itself automatically. If more than one reply is received, the router configures itself according to the most recent reply, accommodating intelligent switches that can handle multiple formats simultaneously.

Using an intelligent retry scheme, LMI autosense attempts to deter mine the LMI type every N391 interval if initially unsuccessful. Default is 60 seconds, which is 6 keep exchanges at 10 seconds each.

To make the typically transparent LMI autosense process visible to the user, turn on "debug frame lmi.” Every N391 interval, the user will se three rapid status queries coming from the serial interface.

Configuring LMI Autosense

No configuration options are provided. You can turn off LMI au tosense by explicitly configuring an LMI type. When the LMI type is written into NVRAM, the next time the router powers up, LMI au tosense is inactive.

Explicitly Configure the LMI

If you want to configure the LMI and thus deactivate LMI autosense, complete the tasks. The tasks in the first two sections are required if you choose to configure the LMI.

- Set the LMI Typ
- Set the LMI Keepalive Interval
- Set the LMI Polling and Timer Intervals

Set the LMI Type

If the router or access server is attached to a public data network (PDN), the LMI type must match the type used on the public network. Otherwise, the LMI type can be set to suit the needs of your privat Frame Relay network.

In interface configuration mode, set one of three types of LMIs—ANSI T1.617 Annex D, Cisco, or ITU-T Q.933 Annex A:

1. Set the LMI type.

   ```
   frame-relay lmi-type {ansi | cisco | q933a}
   ```

2. Write the LMI type to NVRAM. Use the command form that is appropriate to your router platform.

   ```
   copy running-config destination
   ```

Set the LMI Keepalive Interval

A keepalive interval must be set to configure the LMI. By default, this interval is 10 seconds and, per the LMI protocol, must be less than the corresponding interval on the switch.

- To set the keepalive interval, type (in interface configuration mode):

  `keepalive` ***`number`***

- To turn off keepalives on networks without an LMI.

  `no keepalive`

Set the LMI Polling and Timer Intervals

You can set various optional counters, intervals, and thresholds to fine-tune the operation of your LMI DTE and DCE devices. Set these attributes by performing one or more of these tasks in interface configuration mode

- Set the DCE and Network-to-Network Interface (NNI) error threshold.

  `frame-relay lmi-n392dce` ***`threshold`***

- Set the DCE and NNI monitored events count.

  `frame-relay lmi-n393dce` ***`events`***

- Set the polling verification timer on a DCE or NNI interface.

  `frame-relay lmi-t392dce` ***`timer`***

- Set a full status polling interval on a DTE or NNI interface.

  `frame-relay lmi-n391dte` ***`keep-exchanges`***

- Set the DTE or NNI error threshold.

  `frame-relay lmi-n392dte` ***`threshold`***

- Set the DTE and NNI monitored events count.

  `frame-relay lmi-n393dte` ***`events`***

Configure Frame Relay Switched Virtual Circuits

Currently, access to Frame Relay networks is made through private leased lines at speeds ranging from 56 kbps to 45 Mbps. Frame Relay is a connection-oriented, packet-transfer mechanism that establishes virtual circuits between endpoints.

Switched virtual circuits (SVCs) allow access through a Frame Relay network by setting up a path to the destination endpoints only when th need arises and tearing down the path when it is no longer needed.

You must have these services before Frame Relay SVCs can operate:

- Frame Relay SVC support by the service provider—The service provider's switch must be capable of supporting SVC operation.
- Physical loop connection—A leased line or dedicated line must exist between the router (DTE) and the local Frame Relay switch.

You must enable SVC operation at the interface level. Once it is enabled at the interface level, it is enabled on any subinterface on that interface. One signaling channel, DLCI 0, is set up for the interface, and all SVCs are controlled from the physical interface.

To enable Frame Relay SVC service and set up SVCs, complete these tasks. The subinterface tasks are not required, but offer additional flexibility for SVC configuration and operation. The LAPF tasks are not required and not recommended unless you understand thoroughly the impacts on your network.

- Configure SVCs on a Physical Interface
- Configure SVCs on a Subinterface (optional
- Configure a Map Class
- Configure a Map Group with E.164 or X.121 Addresses
- Associate the Map Class with Static Protocol Address Maps
- Configure LAPF Parameters

Configure SVCs on a Physical Interface

To enable SVC operation on a Frame Relay interface, perform these tasks beginning in global configuration mode:

1. Specify the physical interface.

   ```
   interface serial number
   ```

2. Specify the interface IP address, if needed.

   ```
   ip address ip-address mask
   ```

3. Enable Frame Relay encapsulation on the interface.

   encapsulation frame-relay

4. Assign a map group to the interface.

   map-group ***group-name***

5. Enable Frame Relay SVC support on the interface.

   frame-relay svc

Map-group details are specified with the **map-list** command.

Example: SVCs on an Interface

This example configures a physical interface, applies a map-group to the physical interface, and then defines the map-group.

```
interface serial 0
ip address 172.10.8.6
encapsulation frame-relay
map-group bermuda

frame-relay lmi-type q933a
frame-relay svc

map-list bermuda source-addr E164 123456 dest-addr E164 654321
 ip 131.108.177.100 class hawaii
 appletalk 1000.2 class rainbow

map-class frame-relay rainbow
frame-relay idle-timer 60

map-class frame-relay hawaii
frame-relay cir in 64000
frame-relay cir out 64000
```

Configure SVCs on a Subinterface (optional)

To configure Frame Relay SVCs on a subinterface, perform all th tasks in the previous section, except assigning a map group. Then, beginning in global configuration mode, complete these tasks:

1. Specify a subinterface of the main interface configured for SVC operation.

   interface serial ***number.subinterface-number*** **{multipoint | point-to-point}**

2. Specify the subinterface IP address, if needed.

   ip address *ip-address mask*

3. Assign a map group to the subinterface.

   map-group *group-name*

Example: SVCs on a Subinterface

This example configures a point-to-point interface for SVC operation. This example assumes that the main serial 0 interface has been configured for signalling, and that SVC operation has been enabled on th main interface.

```
int s 0.1 point-point
```

Define the map-group; details are specified under the **map-list holiday** command:

```
map-group holiday
```

Associate the map-group with a specific source and destination:

```
map-list holiday local-addr X121 <X121-addr> dest-addr E164 <E164-addr>
```

Specify destination protocol addresses for a map-class:

```
ip 131.108.177.100 class hawaii IETF
 appletalk 1000.2 class rainbow IETF broadcast
```

Define a map class and its QOS settings:

```
map-class hawaii
 frame-relay cir in 2000000
 frame-relay cir out 56000
 frame-relay be 9000
```

Define another map class and its QOS settings:

```
map-class rainbow
 frame-relay cir in 64000
 frame-relay idle-timer 2000
```

Configure a Map Class

Beginning in global configuration mode, configure a map class:

1. Specify the Frame Relay map class name and enter map class configuration mode.

   ```
   map-class frame-relay map-class-name
   ```

2. Specify a custom queue list to be used for the map class.

   ```
   frame-relay custom-queue-list  list-number
   ```

3. Assign a priority queue to virtual circuits associated with the map class.

   ```
   frame-relay priority-group  list-number
   ```

4. Enable BECN feedback to throttle the frame-transmission rate.

   ```
   frame-relay becn-response-enable
   ```

5. Specify the inbound committed information rate (CIR).

   ```
   frame-relay cir in bps
   ```

6. Specify the outbound committed information rate (CIR).

   ```
   frame-relay cir out  bps
   ```

7. Set the minimum acceptable incoming CIR.

   ```
   frame-relay mincir in bps
   ```

8. Set the minimum acceptable outgoing CIR.

   ```
   frame-relay mincir out  bps
   ```

9. Set the incoming committed burst size (Bc).

   ```
   frame-relay bc in bits
   ```

10. Set the outgoing committed burst size (Bc).

    ```
    frame-relay bc out  bits
    ```

11. Set the incoming excess burst size (Be).

    ```
    frame-relay be in bits
    ```

12. Set the outgoing excess burst size (Be).

```
frame-relay be out bits
```

13. Set the idle timeout interval.

```
frame-relay idle-timer duration
```

You can define multiple map classes. A map class is associated with a static map, not with the interface or subinterface. Because of the flexibility this association allows, you can define different map classes fo different destinations.

Configure a Map Group with E.164 or X.121 Addresses

After you have defined a map group for an interface, you can associate the map group with a specific source and destination address, such as E.164 or X.121 addresses. In global configuration mode, specify th map group for a specific interface:

```
map-list group-name source-addr {e164 |
x121} source-address dest-addr {e164 |
x121} destination-address
```

Associate the Map Class with Static Protocol Address Maps

To define the protocol addresses under a **map-list** command and associate each protocol address with a specified map class, use the **class** command for each protocol address. In map class configuration mode, associate a map class with a protocol address:

```
 protocol protocol-address class class-name
[ietf] [broadcast [trigger]]
```

The **ietf** keyword specifies RFC 1490 encapsulation; th **broadcast** keyword specifies that broadcasts must be carried. The **trigger** keyword, which can be configured only if **broadcast** is also configured, enables a broadcast packet to trigger an SVC. If an SVC already exists that uses this map class, the SVC will carry the broad cast.

Configure LAPF Parameters

Frame Relay Link Access Procedure for Frame Relay (LAPF) commands are used to tune Layer 2 system parameters to work well with the Frame Relay switch. Normally, you do not need to change the default settings. However, if the Frame Relay network indicates that it does not support the Frame Reject frame (FRMR) at the LAPF Fram In interface configuration mode, reject procedure:

```
no frame-relay lapf frmr
```

By default, the Frame Reject frame is sent at the LAPF Frame Reject procedure.

Note: ***Manipulation of Layer 2 parameters is not recommended if you do not know well the resulting functional change. For more information, refer to the ITU-T Q.922 specification for LAPF.***

If you must change Layer 2 parameters for your network environment and if you know the resulting functional change, complete these tasks as needed:

- Set the LAPF window size k.

  frame-relay lapf k ***number***

- Set the LAPF maximum retransmission count N200.

  frame-relay lapf n200 ***retries***

- Set the maximum length of the Information field of the LAPF I frame N201.

  frame-relay lapf n201 ***number***

- Set the LAPF retransmission timer value T200.

  frame-relay lapf t200 ***tenths-of-a-second***

- Set the LAPF link idle timer value T203 of DLCI 0.

  frame-relay lapf t203 ***seconds***

Configure Frame Relay Traffic Shaping

Beginning with Release 11.2, AI2524 supports Frame Relay traffic shaping, which provides:

- Rate enforcement on a per-virtual circuit basis—The peak rate for outbound traffic can be set to the CIR or some other user-config ured rate.
- Dynamic traffic throttling on a per-virtual circuit basis—When BECN packets indicate congestion on the network, the outbound traffic rate is automatically stepped down; when congestion eases, the outbound traffic rate is stepped up again. This feature is enabled by default.
- Enhanced queuing support on a per-virtual circuit basis—Either custom queuing or priority queuing can be configured for individual virtual circuits.

By defining separate virtual circuits for different types of traffic and specifying queuing and an outbound traffic rate for each virtual circuit, you can provide guaranteed bandwidth for each type of traffic. By specifying different traffic rates for different virtual circuits over the same line, you can perform virtual time division multiplexing. By throttling outbound traffic from high-speed lines in central offices to lower-speed lines in remote locations, you can ease congestion and data loss in the network; enhanced queuing also prevents congestion-caused data loss.

Traffic shaping applies to both PVCs and SVCs.

To configure Frame Relay traffic shaping, perform these tasks:

- Enable Frame Relay Traffic Shaping on the Interface
- Specify a Traffic-Shaping Map Class for the Interfac
- Define a Map Class with Queuing and Traffic Shaping Parameters
- Define Access Lists
- Define Priority Queue Lists for the Map Class
- Define Custom Queue Lists for the Map Class

Enable Frame Relay Traffic Shaping on the Interface

In interface configuration mode, enable Frame Relay traffic shaping on an interface to enable both traffic shaping and per-virtual circuit queuing on all the interface's PVCs and SVCs:

```
frame-relay traffic-shaping
```

Specify a Traffic-Shaping Map Class for the Interface

If you specify a Frame Relay map class for a main interface, all the virtual circuits on its subinterfaces inherit all the traffic shaping parame ters defined for the class.

In interface configuration mode, specify a map class for the specified interface:

```
frame-relay class map-class-name
```

You can override the default for a specific DLCI on a specific subinterface by using the **class virtual circuit** configuration command to assign the DLCI explicitly to a different class.

Define a Map Class with Queuing and Traffic Shaping Parameters

When you define a map class for Frame Relay, you can define the average and peak rates (in bits per second) allowed on virtual circuits associated with the map class. You can also specify either a custom queue-list or a priority queue-group to use on virtual circuits associated with the map class (optional).

Beginning in global configuration mode, define a map class:

1. Specify a map class to define.

   `map-class frame-relay` *`map-class-name`*

2. Define the traffic rate for the map class.

   `frame-relay traffic-rate` *`average`* **`[`***`peak`***`]`**

3. Specify a custom queue-list.

   `frame-relay custom-queue-list` *`number`*

4. Specify a priority queue-list.

   `frame-relay priority-group` *`number`*

Define Access Lists

You can specify access lists and associate them with the custom queue-list defined for any map class. The list number specified in th access list and the custom queue list tie them together.

See the appropriate protocol chapters for information about defining access lists for the protocols you want to transmit on the Frame Relay network.

Define Priority Queue Lists for the Map Class

You can define a priority list for a protocol and a default priority list. The number used for a specific priority list ties the list to the Frame Relay priority group defined for a specified map class.

For example, if you enter th **`frame relay priority-group 2`** command for the map class fast_vcs and then you enter th **`priority-list 2 protocol decnet high`** command, that priority list is used for the fast_vcs map class. The average and peak traffic rates defined for the fast_vcs map class are used for traffic.

Define Custom Queue Lists for the Map Class

You can define a queue list for a protocol and a default queue list. You can also specify the maximum number of bytes to be transmitted in any cycle. The number used for a specific queue list ties the list to the Frame Relay custom-queue list defined for a specified map class.

For example, if you enter the **`frame relay custom-queue-list 1`** command for the map class slow_vcs and then you enter th **`queue-list 1 protocol ip list 100`** command, that queue list is used for the slow_vcs map class; **`access-list 100`** definition is also used for that map class and queue. The average and peak traffic rates defined for the slow_vcs map class are used for IP traffi that meets the access list 100 criteria.

Example: Frame Relay Traffic Shaping

This example illustrates a Frame Relay interface with three point-to point subinterfaces.

In this example, the virtual circuits on subinterfaces Serial0.1 a Serial0.2 inherit class parameters from the main interface, namely those defined in slow_vcs. However, the virtual circuit defined on subinterface Serial0.2 (DLCI 102) is specifically configured to use map class fast_vcs.

Map class slow_vcs uses a peak rate of 9600 and average rate of 4800 bps. Because BECN feedback is enabled by default, the output rate will be cut back as low as 4800bps in response to received BECNs. This map class is configured to use custom queuing using queue-list 1. In this example, queue-list 1 has 3 queues, with the first two being controlled by access lists 100 and 115.

Map class fast_vcs uses a peak rate of 64000 and average rate of 16000 bps. Because BECN feedback is enabled by default, the output rate will be cut back as low as 4800bps in response to received BECNs. This map class is configured to use priority-queuing using priority-group 2.

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi
 frame-relay traffic-shaping
 frame-relay class slow_vcs
!
interface Serial0.1 point-to-point
 ip address 10.128.30.1 255.255.255.248
 ip ospf cost 200
 bandwidth 10
 frame-relay interface-dlci 101
!
interface Serial0.2 point-to-point
 ip address 10.128.30.9 255.255.255.248
 ip ospf cost 400
 bandwidth 10
 frame-relay interface-dlci 102
 class fast_vcs
!
interface Serial0.3 point-to-point
 ip address 10.128.30.17 255.255.255.248
 ip ospf cost 200
 bandwidth 10
 frame-relay interface-dlci 103
!
map-class frame-relay slow_vcs
 frame-relay traffic-rate 4800 9600
 frame-relay custom-queue-list 1
!
map-class frame-relay fast_vcs
 frame-relay traffic-rate 16000 64000
 frame-relay priority-group 2
!
access-list 100 permit tcp any any eq 2065
access-list 115 permit tcp any any eq 256
!
priority-list 2 protocol decnet hig
priority-list 2 ip nor
priority-list 2 default mediu
!
queue-list 1 protocol ip 1 list 100
queue-list 1 protocol ip 2 list 115
queue-list 1 default 3
queue-list 1 queue 1 byte-count 1600 limit 200
queue-list 1 queue 2 byte-count 600 limit 200
queue-list 1 queue 3 byte-count 500 limit 200
```

Customize Frame Relay for Your Network

Perform these tasks to customize Frame Relay:

- Configure Frame Relay Subinterfaces
- Configure Frame Relay Switching
- Disable or Reenable Frame Relay Inverse ARP
- Create a Broadcast Queue for an Interface
- Configure Payload Compression
- Configure TCP/IP Header Compression
- Configure Discard Eligibility
- Configure DLCI Priority Levels

Configure Frame Relay Subinterfaces

To understand and define Frame Relay Subinterfaces, perform these tasks:

- Understand Frame Relay Subinterfaces
- Define Frame Relay Subinterfaces
- Define Subinterface Addressing

After these tasks are completed, you can also perform these optional tasks:

- Configure Transparent Bridging for Frame Relay
- Configure a Backup Interface for a Subinterface

Understand Frame Relay Subinterfaces

Frame Relay subinterfaces provide a mechanism for supporting partially meshed Frame Relay networks. Most protocols assume transitivity on a logical network; that is, if station A can talk to station B, and station B can talk to station C, then station A should be able to talk to station C directly. Transitivity is true on LANs, but not on Fram Relay networks unless A is directly connected to C.

Additionally, certain protocols such as transparent bridging cannot b supported on partially meshed networks because they require "split horizon," in which a packet received on an interface cannot be trans mitted out the same interface even if the packet is received and trans mitted on different virtual circuits.

Configuring Frame Relay subinterfaces ensures that a single physical interface is treated as multiple virtual interfaces. This capability allows us to overcome split horizon rules. Packets received on one virtual interface can now be forwarded out another virtual interface, even if they are configured on the same physical interface.

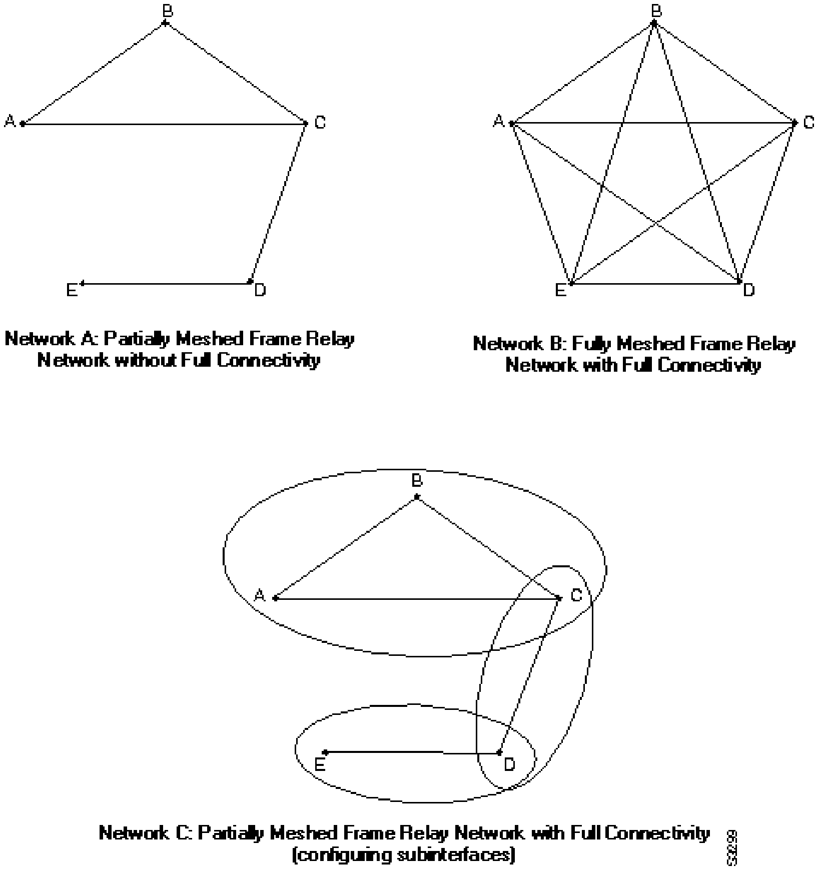
Subinterfaces address the limitations of Frame Relay networks by providing a way to subdivide a partially meshed Frame Relay network into a number of smaller, fully meshed (or point-to-point) subnet works. Each subnetwork is assigned its own network number and appears to the protocols as if it is reachable through a separate interface.

Note: ***Point-to-point subinterfaces can be unnumbered for us with IP, reducing the addressing burden that might otherwise result.***

For example, suppose you have a 5-node Frame Relay network (see Figure 13-2) that is partially meshed (Network A). If the entire network is viewed as a single subnetwork (with a single network number assigned), most protocols assume that node A can transmit a packet directly to node E, when in fact it must be relayed through nodes C and D. This network can be made to work with certain protocols (for example, IP) but will not work at all with other protocols because nodes C and D will not relay the packet out the same interface on which it was received. One way to make this network function fully is to creat a fully meshed network (Network B), but doing so requires a larg number of PVCs, which may not be economically feasible.

Using subinterfaces, you can subdivide the Frame Relay network into three smaller subnetworks (Network C) with separate network numbers. Nodes A, B, and C are connected to a fully meshed network, and nodes C and D, as well as nodes D and E are connected via point-to point networks. In this configuration, nodes C and D can access two subinterfaces and can therefore forward packets without violating split horizon rules. If transparent bridging is being used, each subinterfac is viewed as a separate bridge port.

Figure 13-2:Using Subinterfaces to Provide Full Connectivity on a Partially Meshed Frame Relay Network



Network A: Partially Meshed Frame Relay Network without Full Connectivity

Network B: Fully Meshed Frame Relay Network with Full Connectivity

Network C: Partially Meshed Frame Relay Network with Full Connectivity (configuring subinterfaces)

Define Frame Relay Subinterfaces

Configure subinterfaces on a Frame Relay network:

1. Specify a serial interface.

   `interface serial` ***`number`***

2. Configure Frame Relay encapsulation on the serial interface.

   `encapsulation frame-relay`

3. Specify a subinterface.

   `interface serial` ***`number.subinterface-number`***
   `{multipoint | point-to-point}`

Subinterfaces can be configured for multipoint or point-to-point communication. There is no default.

Define Subinterface Addressing

For point-to-point subinterfaces, the destination is presumed to be known and is identified or implied in the **`frame-relay interface-dlci`** command. For multipoint subinterfaces, the destinations can be dynamically resolved through the use of Frame Relay Inverse ARP or can be statically mapped through the use of the **`frame-relay map`** command.

Addressing on Point-to-Point Subinterfaces

If you specified a point-to-point subinterface in Step 3 unde Define Frame Relay Subinterfaces, type (in interface configuration mode):

`frame-relay interface-dlci` ***`dlci`*** **`[`*`option`*`]`**

If you define a subinterface for point-to-point communication, you cannot reassign the same subinterface number to be used for multipoint communication without first rebooting the router or access server. Instead, you can simply avoid using that subinterface numbe and use a different subinterface number instead.

Examples: Basic Subinterface

In this example, subinterface 1 models a point-to-point subnet and subinterface 2 models a broadcast subnet. For emphasis, the multipoint keyword is used for serial subinterface 2, even though a subinterface is multipoint by default.

```
interface serial 0
encapsulation frame-relay
interface serial 0.1 point-to-point
ip address 10.0.1.1 255.255.255.0
frame-relay interface-dlci 42

interface serial 0.2 multipoint
ip address 10.0.2.1 255.255.255.0
frame-relay map 10.0.2.2 18
```

Addressing on Multipoint Subinterfaces

If you specified a multipoint subinterface in Step 3 under Define Frame Relay Subinterfaces, perform the tasks in one or both of the these sections:

- Accept Inverse ARP for Dynamic Address Mapping on Multipoint Subinterfaces
- Configure Static Address Mapping on Multipoint Subinterfaces

You can configure some protocols for dynamic address mapping and others for static address mapping.

Accept Inverse ARP for Dynamic Address Mapping on Multipoint Subinterfaces

Dynamic address mapping uses Frame Relay Inverse ARP to request the next hop protocol address for a specific connection, given a DLCI. Responses to Inverse ARP requests are entered in an address-to-DLC mapping table on the router or access server; the table is then used to supply the next hop protocol address or the DLCI for outgoing traffic.

Since the physical interface is now configured as multiple subinterfaces, you must provide information that distinguishes a subinterface from the physical interface and associates a specific subinterface with a specific DLCI.

In interface configuration mode, associate a specific multipoint subinterface with a specific DLCI:

```
frame-relay interface-dlci  dlci
```

Inverse ARP is enabled by default for all protocols it supports, but can be disabled for specific protocol-DLCI pairs. As a result, you can use dynamic mapping for some protocols and static mapping for other protocols on the same DLCI. You can explicitly disable Inverse ARP for a protocol-DLCI pair if you know the protocol is not supported on th other end of the connection.

Because Inverse ARP is enabled by default for all protocols that it supports, no additional command is required to configure dynamic address mapping on a subinterface.

Example: Frame Relay Multipoint Subinterface with Dynamic Addressing

This example configures two multipoint subinterfaces for dynamic address resolution. Each subinterface is provided with an individual protocol address and subnet mask, and the **`interface-dlci`** command associates the subinterface with a specified DLCI. Addresses of remote destinations for each multipoint subinterface will b resolved dynamically.

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi
!
interface Serial0.103 multipoint
 ip address 172.21.177.18 255.255.255.0
 frame-relay interface-dlci 300
!
interface Serial0.104 multipoint
 ip address 172.21.178.18 255.255.255.0
 frame-relay interface-dlci 400
```

Configure Static Address Mapping on Multipoint Subinterfaces

A static map links a specified next hop protocol address to a specified DLCI. Static mapping removes the need for Inverse ARP requests; when you supply a static map, Inverse ARP is automatically disabled for the specified protocol on the specified DLCI.

You must use static mapping if the router at the other end either does not support Inverse ARP at all or does not support Inverse ARP for a specific protocol that you want to use over Frame Relay.

To establish static mapping according to your network needs, perform one of these tasks in interface configuration mode:

- Define the mapping between a next hop protocol address and the DLCI used to connect to the address.

  frame-relay map ***protocol protocol-address dlci*** **[broadcast] [ietf] [cisco]**

- Define a DLCI used to send ISO CLNS frames.

  frame-relay map clns ***dlci*** **[broadcast]**

- Define a DLCI used to connect to a bridge.

  frame-relay map bridge ***dlci*** **[ietf] broadcast**

Use these keywords to specify the protocols:

- **ip**—IP
- **ipx**—Novell IPX
- **clns**—ISO CLNS

The **broadcast** keyword is required for routing protocols such as OSI protocols and the Open Shortest Path First (OSPF) protocol. Se the **frame-relay map** command description in the Wide-Area Networking Command Reference for more information about using the **broadcast** keyword.

Example: IPX Routes over Frame Relay Subinterfaces

This example configures a serial interface for Frame Relay encapsulation and sets up multiple IPX virtual networks corresponding to Frame Relay subinterfaces:

```
ipx routing 0000.0c02.5f4f
!
interface serial 0
encapsulation frame-relay
interface serial 0.1 multipoint
ipx network 1
frame-relay map ipx 1.000.0c07.d530 200 broadcast
ipx network 2
frame-relay map ipx 2.000.0c07.d530 300 broadcast
```

For subinterface serial 0.1, the router at the other end might be configured as:

```
ipx routing
interface serial 2 multipoint
ipx network 1
frame-relay map ipx 1.000.0c02.5f4f 200 broadcast
```

Configure Transparent Bridging for Frame Relay

Transparent bridging for Frame Relay encapsulated serial and HSSI interfaces is supported on the AI2524 router. Transparent bridging for Frame Relay encapsulated serial interfaces is supported on our access servers.

You can configure transparent bridging for point-to-point or point-to-multipoint subinterfaces.

Note: ***All PVCs configured on a subinterface belong to the same bridge group.***

Point-to-Point Subinterfaces

In interface configuration mode, configure transparent bridging for point-to-point subinterfaces:

1. Specify a serial interface.

   `interface serial` ***`number`***

2. Configure Frame Relay encapsulation on the serial interface.

   `encapsulation frame-relay`

3. Specify a subinterface.

   `interface serial` ***`number.subinterface-number`*** **`point-to-point`**

4. Associate a DLCI with the subinterface.

   `frame-relay interface-dlci` ***`dlci`*** **`[`*****`option`*****`]`**

5. Associate the subinterface with a bridge group.

   `bridge-group` ***`bridge-group`***

Example: Unnumbered IP over a Point-to-Point Subinterface

This example sets up unnumbered IP over subinterfaces at both ends of a point-to-point connection. In this example, Router A functions as the DTE, and Router B functions as the DCE. Routers A and B are both attached to Token Ring networks.

Configuration for Router A

```
frame-relay switching
!
interface token-ring 0
ip address 131.108.177.1 255.255.255.0
!
interface serial 0
no ip address
encapsulation frame-relay IETF
!

interface Serial0.2 point-to-point
ip unnumbered TokenRing0
ip pim sparse-mode
frame-relay interface-dlci 20
```

Configuration for Router B

```
frame-relay switching
!
interface token-ring 0
ip address 131.108.178.1 255.255.255.0
!
interface serial 0
no ip address
encapsulation frame-relay IETF
bandwidth 384
clockrate 4000000
frame-relay intf-type dce
!
interface serial 0.2 point-to-point
ip unnumbered TokenRing1
ip pim sparse-mode

bandwidth 384
frame-relay interface-dlci 20
```

Point-to-Multipoint Interfaces

In interface configuration mode, configure transparent bridging for point-to-multipoint subinterfaces:

1. Specify a serial interface.

   interface serial ***number***

2. Configure Frame Relay encapsulation on the serial interface.

   encapsulation frame-relay

3. Specify a subinterface.

   interface serial ***number.subinterface-number*** **multipoint**

4. Define the mapping between a next hop protocol address and the DLCI used to connect to the address.

   frame-relay map bridge ***dlci*** **[broadcast] [ietf]**

5. Associate the subinterface with a bridge group.

   bridge-group ***bridge-group***

Example: Transparent Bridging Using Subinterfaces

In this example, Frame Relay DLCIs 42, 64, and 73 are to be used as separate point-to-point links with transparent bridging running over them. The bridging spanning tree algorithm views each PVC as a separate bridge port, and a frame arriving on the PVC can be relayed back out a separate PVC. Be sure that routing is not enabled when configuring transparent bridging using subinterfaces.

```
interface serial 0
encapsulation frame-relay
interface serial 0.1 point-to-point
bridge-group 1
frame-relay interface-dlci 42
interface serial 0.2 point-to-point
bridge-group 1
frame-relay interface-dlci 64
interface serial 0.3 point-to-point
bridge-group 1
frame-relay interface-dlci 73
```

Configure a Backup Interface for a Subinterface

Both point-to-point and multipoint Frame Relay subinterfaces can be configured with a backup interface. This approach allows individual PVCs to be backed up in case of failure rather than depending on th entire Frame Relay connection to fail before the backup takes over. You can configure a subinterface for backup on failure only, not for backup based on loading of the line.

If the serial interface has a backup interface, it will have precedenc over the subinterface's backup interface in the case of complete loss o connectivity with the Frame Relay network. As a result, a subinterface backup is activated only if the serial interface is up, or if the serial interface is down and does not have a backup interface defined. If a subinterface has failed while its backup is in use, and then the serial interface goes down, the subinterface backup stays connected.

Beginning in global configuration mode, configure a backup interface for a Frame Relay subinterface:

1. Specify the interface.

   interface serial ***number***

2. Configure Frame Relay encapsulation.

   encapsulation frame-relay

3. Configure the subinterface.

   ```
   interface serial number.subinterface-number
   point-to-point
   ```

4. Specify a DLCI for the subinterface.

   ```
   frame-relay interface-dlci  dlci
   ```

5. Specify a backup interface for the subinterface.

   ```
   backup interface serial  number
   ```

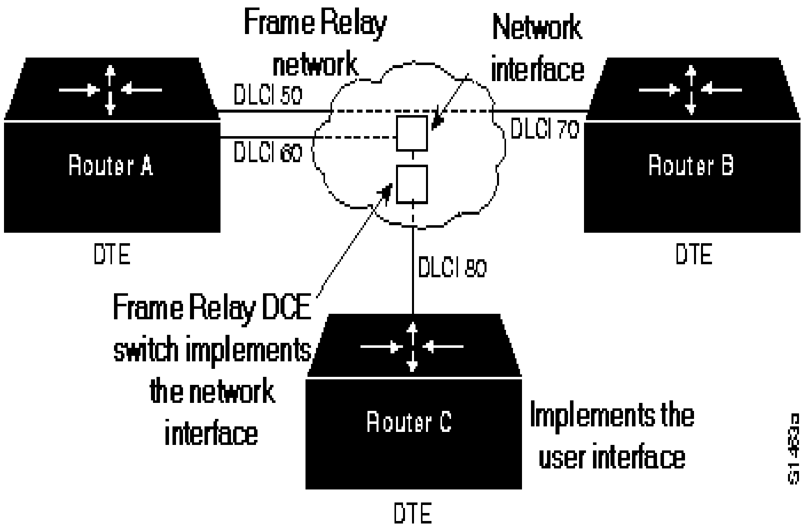
6. Specify backup enable and disable delay.

   ```
   backup delay {enable-delay | disable-delay}
   ```

Configure Frame Relay Switching

Frame Relay switching is a means of switching packets based upon the DLCI, which can be looked upon as the Frame Relay equivalent of a MAC address. You perform the switching by configuring your router or access server as a Frame Relay network. There are two parts to a Frame Relay network: a Frame Relay DTE (the router or access server) and a Frame Relay DCE switch. Figure 13-3 illustrates this concept.

Figure 13-3:Frame Relay Switched Network



In Figure 13-3, Routers A, B, and C are Frame Relay DTEs connected to each other via a Frame Relay network. Our implementation of Frame Relay switching allows our devices to be used as depicted in this Frame Relay network.

Perform the tasks in these sections, as necessary, to configure Frame Relay switching:

- Enable Frame Relay Switching
- Configure a Frame Relay DTE Device, DCE Switch, or NNI Support
- Specify the Static Rout

Enable Frame Relay Switching

In global configuration mode, enable packet switching before configuring it on a Frame Relay DTE or DCE or with Network-to-Network Interface (NNI) support:

```
frame-relay switching
```

Configure a Frame Relay DTE Device, DCE Switch, or NNI Support

In interface configuration mode, configure an interface as a DTE device or a DCE switch or as a switch connected to a switch to support NNI connections:

```
frame-relay intf-type [dce | dte | nni]
```

DCE is the default.

Specify the Static Route

In interface configuration mode, specify a static route for PVC switching:

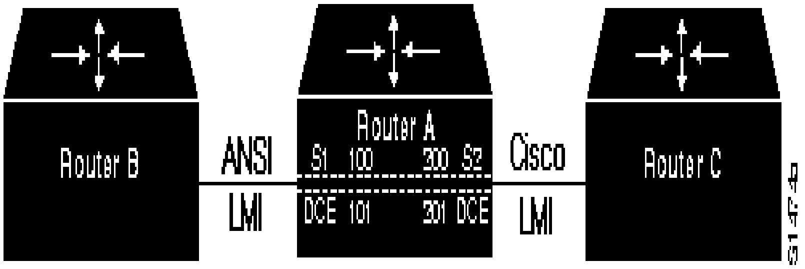
```
frame-relay route in-dlci out-interface
out-dlci
```

Example: PVC Switching Configuration

You can configure your router as a dedicated, DCE-only Frame Relay switch. Switching is based on DLCIs. The incoming DLCI is examined, and the outgoing interface and DLCI are determined. Switching takes place when the incoming DLCI in the packet is replaced by the outgoing DLCI, and the packet is sent out the outgoing interface.

In this example, the router switches two PVCs between interface serial 1 and 2. Frames with DLCI 100 received on serial 1 will be transmitted with DLCI 200 on serial 2 (see Figure 13-4).

Figure 13-4:PVC Switching Configuration



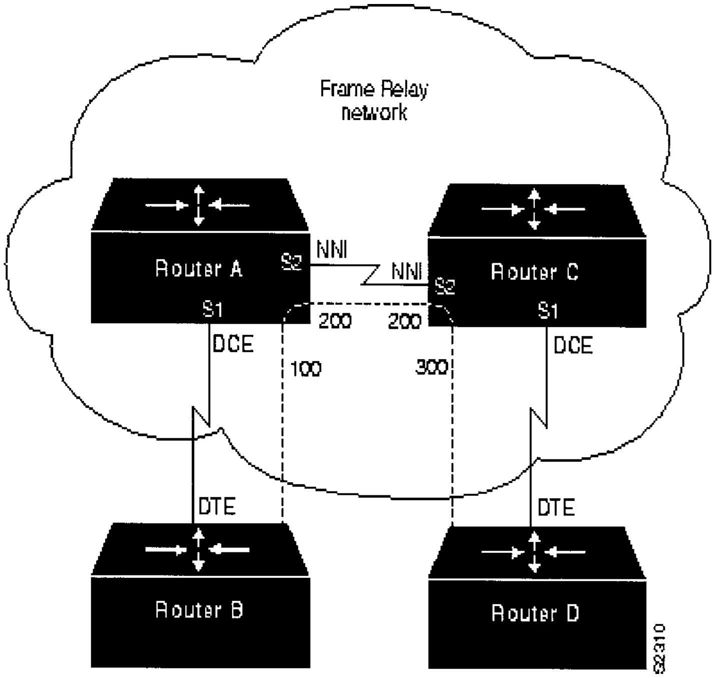
Configuration for Router A

```
frame-relay switching
!
interface Ethernet0
ip address 131.108.160.58 255.255.255.0
!
interface Serial1
no ip address
encapsulation frame-relay
keepalive 15
frame-relay lmi-type ansi
frame-relay intf-type dce
frame-relay route 100 interface Serial2 200
frame-relay route 101 interface Serial2 201
clockrate 2000000
!
interface Serial2
encapsulation frame-relay keepalive 15
frame-relay intf-type dce
frame-relay route 200 interface Serial1 100
frame-relay route 201 interface Serial1 101
clockrate 64000
```

Example: Pure Frame Relay DCE

Using the PVC switching feature, it is possible to build an entire Frame Relay network using our routers. In this example, Router A and Router C act as Frame Relay switches implementing a two-node network. Th standard Network-to-Network Interface (NNI) signaling protocol is used between Router A and Router C (see Figure 13-5).

Figure 13-5:Frame Relay DCE Configuration



Configuration for Router A

```
frame-relay switching
!
interface ethernet 0
no ip address
shutdown
!
interface ethernet 1
no ip address
shutdown
!
interface ethernet 2
no ip address
shutdown
!
interface ethernet 3
no ip address
shutdown
!
interface serial 0
ip address 131.108.178.48 255.255.255.0
shutdown
!
interface serial 1
no ip address
encapsulation frame-relay
frame-relay intf-type dce
frame-relay lmi-type ansi
frame-relay route 100 interface serial 2 200
!
interface serial 2
no ip address
encapsulation frame-relay
frame-relay intf-type nni

frame-relay lmi-type q933a
frame-relay route 200 interface serial 1 100
clockrate 2048000
!
interface serial 3
no ip address
shutdown
```

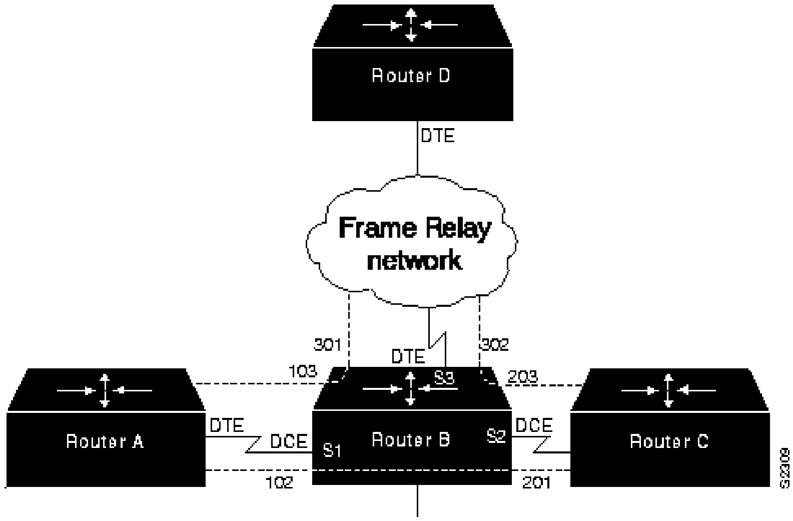
Configuration for Router C

```
frame-relay switching
!
interface ethernet 0
no ip address
shutdown
!
interface ethernet1
no ip address
shutdown
!
interface ethernet 2
no ip address
shutdown
!
interface ethernet 3
no ip address
shutdown
!
interface serial 0
ip address 131.108.187.84 255.255.255.0
shutdown
!
interface serial 1
no ip address
encapsulation frame-relay
frame-relay intf-type dce
frame-relay route 300 interface serial 2 200
!
interface serial 2
no ip address
encapsulation frame-relay
frame-relay intf-type nni
frame-relay lmi-type q933a
frame-relay route 200 interface serial 1 300
!
interface serial 3
no ip address
shutdown
```

Example: Hybrid DTE/DCE PVC Switching

Routers can also be configured as hybrid DTE/DCE Frame Relay switches (see Figure 13-6).

Figure 13-6:Hybrid DTE/DCE PVC Switching



In this example, Router B acts as a hybrid DTE/DCE Frame Relay switch. It can switch frames between the two DCE ports and between a DCE port and a DTE port. Traffic from the Frame Relay network can also be terminated locally. In the example, three PVCs are defined as:

- Serial 1, DLCI 102 to serial 2, DLCI 201—DCE switching
- Serial 1, DLCI 103 to serial 3, DLCI 301—DCE/DTE switching
- Serial 2, DLCI 203 to serial 3, DLCI 302—DCE/DTE switching

DLCI 400 is also defined for locally terminated traffic.

Configuration for Router B

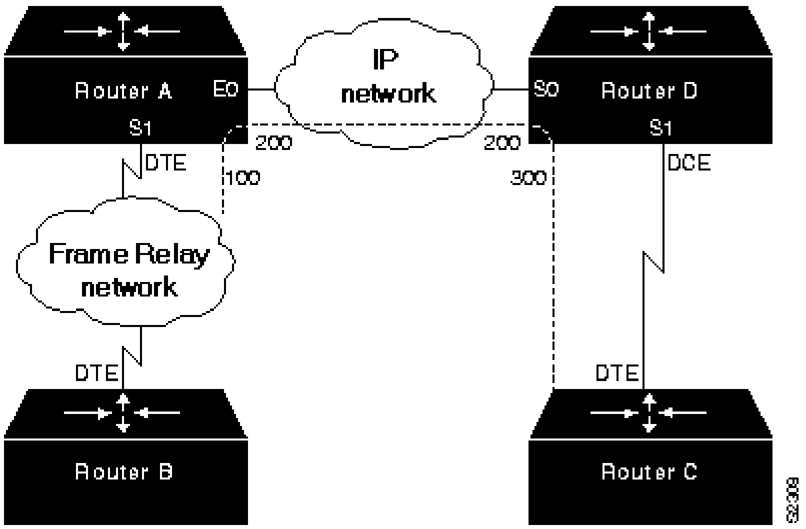
```
frame-relay switching
!
interface ethernet 0
ip address 131.108.123.231 255.255.255.0
!
interface ethernet 1
ip address 131.108.5.231 255.255.255.0
!
interface serial 0
no ip address
shutdown
!
interface serial 1
no ip address
encapsulation frame-relay
frame-relay intf-type dce

frame-relay route 102 interface serial 2 201
frame-relay route 103 interface serial 3 301
!
interface serial 2
no ip address
encapsulation frame-relay
frame-relay intf-type dce
frame-relay route 201 interface serial 1 102
frame-relay route 203 interface serial 3 302
!
interface serial 3
ip address 131.108.111.231
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay route 301 interface serial 1 103
frame-relay route 302 interface serial 1 203
frame-relay map ip 131.108.111.4 400 broadcast
```

Example: Switching over an IP Tunnel

You can switch over an IP tunnel by creating a point-to-point tunnel across the internetwork over which PVC switching can take place (se Figure 13-7).

Figure 13-7:Frame Relay Switch over IP Tunnel



The following configurations illustrate how to create the IP network depicted in Figure 13-7.

Configuration for Router A

```
frame-relay switching
!
interface Ethernet0
ip address 108.131.123.231 255.255.255.0
!
interface Ethernet1
ip address 131.108.5.231 255.255.255.0
!
interface Serial0
no ip address
shutdown
!
interface Serial1
ip address 131.108.222.231 255.255.255.0
encapsulation frame-relay
frame-relay map ip 131.108.222.4 400 broadcast
frame-relay route 100 interface Tunnel1 200
!
interface Tunnel1
tunnel source Ethernet0
tunnel destination 150.150.150.123
```

Configuration for Router D

```
frame-relay switching
!
interface Ethernet0
ip address 131.108.231.123 255.255.255.0
!
interface Ethernet1
ip address 131.108.6.123 255.255.255.0
!
interface Serial0
ip address 150.150.150.123 255.255.255.0
encapsulation ppp
interface Tunnel1
tunnel source Serial0
tunnel destination 108.131.123.231
!
interface Serial1
ip address 131.108.7.123 255.255.255.0
encapsulation frame-relay
frame-relay intf-type dce
frame-relay route 300 interface Tunnel1 200
```

Disable or Reenable Frame Relay Inverse ARP

Frame Relay Inverse ARP is a method of building dynamic address mappings in Frame Relay networks running IP and Novell IPX. Inverse ARP allows the router or access server to discover the protocol address of a device associated with the virtual circuit.

Inverse ARP creates dynamic address mappings, as contrasted with the **`frame-relay map`** command, which defines static mappings between a specific protocol address and a specific DLCI.

Inverse ARP is enabled by default but can be disabled explicitly for a given protocol and DLCI pair. Disable or reenable Inverse ARP under these conditions:

- Disable Inverse ARP for a selected protocol and DLCI pair when you know that the protocol is not supported on the other end of th connection.
- Reenable Inverse ARP for a protocol and DLCI pair if conditions or equipment change and the protocol is then supported on the other end of the connection.

Note: ***If you change from a point-to-point subinterface to a multipoint subinterface, then change the subinterface number. Frame Relay Inverse ARP will be on by default, and no further action is required.***

You do not need to enable or disable Inverse ARP if you have a point-to-point interface, because there is only a single destination and discovery is not required.

To select Inverse ARP or disable it, perform one of these tasks in interface configuration mode:

- Enable Frame Relay Inverse ARP for a specific protocol and DLCI pair, only if it was previously disabled.

  ```
  frame-relay inverse-arp  protocol dlci
  ```

- Disable Frame Relay Inverse ARP for a specific protocol and DLCI pair.

  ```
  no frame relay inverse-arp  protocol dlci
  ```

Create a Broadcast Queue for an Interface

Very large Frame Relay networks might have performance problems when many DLCIs terminate in a single router or access server that must replicate routing updates and service advertising updates on each DLCI. The updates can consume access-link bandwidth and cause significant latency variations in user traffic; the updates can also consum interface buffers and lead to higher packet rate loss for both user dat and routing updates.

To avoid such problems, you can create a special broadcast queue for an interface. The broadcast queue is managed independently of th normal interface queue, has its own buffers, and has a configurable size and service rate.

A broadcast queue is given a maximum transmission rate (throughput) limit measured in both bytes per second and packets per second. The queue is serviced to ensure that no more than this maximum is pro vided. The broadcast queue has priority when transmitting at a rate below the configured maximum, and hence has a guaranteed minimum bandwidth allocation. The two transmission rate limits are intended to avoid flooding the interface with broadcasts. The actual transmission rate limit in any second is the first of the two rate limits that is reached.

In interface configuration mode, create a broadcast queue:

```
frame-relay broadcast-queue  size byte-rate
packet-rate
```

Configure Payload Compression

You can configure payload compression on point-to-point or multi point interfaces or subinterfaces. Payload compression uses the stac method to predict what the next character in the frame will be. Becaus the prediction is done packet-by-packet, the dictionary is not con served across packet boundaries.

Payload compression on each virtual circuit consumes approximately 40 kilobytes for dictionary memory.

- Configure payload compression on a specified multipoint interface or subinterface:

  ```
  frame-relay map protocol protocol-address
  dlci
  ```

  ```
  payload-compress packet-by-packet
  ```

- Configure payload compression on a specified point-to-point interface or subinterface:

  ```
  frame-relay payload-compress packet-by-
  packet
  ```

Configure TCP/IP Header Compression

TCP/IP header compression, as described by RFC 1144, is designed to improve the efficiency of bandwidth use over low-speed serial links.

A typical TCP/IP packet includes a 40-byte datagram header. Once a connection is established, the header information need not be repeated in every packet that is sent. Reconstructing a smaller header that identifies the connection and indicates the fields that changed and the amount of change reduces the number of bytes transmitted. The average compressed header is 10 bytes long.

For this algorithm to function, packets must arrive in order. If packets arrive out of order, the reconstruction will appear to create regular TCP/IP packets but the packets will not match the original.

Note: ***Because priority queuing changes the order in which packets are transmitted, enabling priority queueing on th interface is not recommended.***

You can configure TCP/IP header compression in either of two ways:

- Configure an Individual IP Map for TCP/IP Header Compression
- Configure an Interface for TCP/IP Header Compression

Note: ***If you configure an interface with Cisco encapsulation and TCP/IP header compression, Frame Relay IP maps inherit the compression characteristics of the interface. However, if you configure the interface with IETF encapsulation, the interface cannot be configured for compression. Frame Relay maps will have to b configured individually to support TCP/IP header compression.***

Configure an Individual IP Map for TCP/IP Header Compression

TCP/IP header compression requires Cisco encapsulation. If you need to have IETF encapsulation on an interface as a whole, you can still configure a specific IP map to use Cisco encapsulation and TCP header compression.

In addition, even if you configure the interface to perform TCP/IP header compression, you can still configure a specific IP map not to compress TCP/IP headers.

You can specify whether TCP/IP header compression is active or passive. Active compression subjects every outgoing packet to TCP/IP header compression. Passive compression subjects an outgoing TCP/IP packet to header compression only if the packet had a compressed TCP/IP header when it was received.

In interface configuration mode, configure an IP map to use Cisco encapsulation and TCP/IP header compression:

```
frame-relay map ip ip-address dlci
[broadcast] cisco tcp headercompression
{active | passive}
```

The default encapsulation is cisco.

Note: ***An interface that is configured to support TCP/IP header compression cannot also support priority queuing or custom queuing.***

Configure an Interface for TCP/IP Header Compression

You can configure the interface with active or passive TCP/IP header compression. Active compression, the default, subjects all outgoing TCP/IP packets to header compression. Passive compression subjects an outgoing packet to header compression only if the packet had a compressed TCP/IP header when it was received on that interface.

In interface configuration mode, apply TCP/IP header compression to an interface:

1. Configure Cisco encapsulation on the interface.

```
encapsulation frame-relay
```

2. Enable TCP/IP header compression on the interface.

```
 frame-relay ip tcp header-compression
[passive]
```

Note: ***If an interface configured with Cisco encapsulation is later configured with IETF encapsulation, all TCP/IP header compression characteristics are lost. To appl TCP/IP header compression over an interface configured with IETF encapsulation, you must configure individual IP maps, as described in the section Configure an Individual IP Map for TCP/IP Header Compression.***

Example: IP Map with Inherited TCP/IP Header Compression

This example shows an interface configured for TCP/IP header compression and an IP map that inherits the compression characteristics. Note that the Frame Relay IP map is not explicitly configured for header compression.

```
interface serial 1
encapsulation frame-relay
ip address 131.108.177.178 255.255.255.0
frame-relay map ip 131.108.177.177 177 broadcast
frame-relay ip tcp header-compression passive
```

Use the **`show frame-relay map`** command to display the resulting compression and encapsulation characteristics; the IP map has inherited passive TCP/IP header compression:

```
Router> show frame-relay map
Serial 1(administratively down): ip 131.108.177.177
dlci 177 (0xB1,0x2C10), static,
broadcast,
CISCO
TCP/IP Header Compression (inherited), passive (inherited
)
```

Example: Using an IP Map to Override TCP/IP Header Compression

This example shows the use of a Frame Relay IP map to override the compression set on the interface:

```
interface serial 1
encapsulation frame-relay
ip address 131.108.177.178 255.255.255.0
frame-relay map ip 131.108.177.177 177 broadcast nocompress
frame-relay ip tcp header-compression passive
```

Use the **`show frame-relay map`** command to display the resulting compression and encapsulation characteristics; the IP map has not inherited TCP header compression:

```
Serial 1 (administratively down): ip 131.108.177.177
dlci 177 (0xB1,0x2C10), static,
broadcast, CISCO
```

Disable TCP/IP Header Compression

You can disable TCP/IP header compression by using either of two commands that have different effects, depending on whether Fram Relay IP maps have been explicitly configured for TCP/IP header compression or have inherited their compression characteristics from the interface.

Frame Relay IP maps that have explicitly configured TCP/IP heade compression must also have TCP/IP header compression explicitly disabled.

To disable TCP/IP header compression, perform one of these tasks in interface configuration mode:

- Disable TCP/IP header compression on all Frame Relay IP maps that are not explicitly configured for TCP header compression.

  ```
  no frame-relay ip tcp header-compression
  ```

  ```
  frame-relay map ip ip-address dlci
  ```

- Disable TCP/IP header compression on a specified Frame Relay IP map.

  ```
  nocompress tcp header-compression
  ```

Example: Disabling Inherited TCP/IP Header Compression

In this first example, the initial configuration is:

```
interface serial 1
encapsulation frame-relay
ip address 131.108.177.179 255.255.255.0
frame-relay ip tcp header-compression passive
frame-relay map ip 131.108.177.177 177 broadcast
frame-relay map ip 131.108.177.178 178 broadcast tcp header-compression
```

Enter these commands:

```
serial interface 1
no frame-relay ip tcp header-compression
```

Use the **show frame-relay map** command to display the resulting compression and encapsulation characteristics:

```
Router> show frame-relay map
Serial 1 (administratively down): ip 131.108.177.177 177
dlci 177(0xB1, 0x2C10), static,
broadcast
CISCO
Serial 1 (administratively down): ip 131.108.177.178 178
dlci 178(0xB2,0x2C20), static
broadcast
CISCO
TCP/IP Header Compression (enabled)
```

As a result, header compression is disabled for the first map (with DLCI 177), which inherited its header compression characteristics from the interface. However, header compression is not disabled for the second map (DLCI 178), which is explicitly configured for header compression.

Example: Disabling Explicit TCP/IP Header Compression

In this second example, the initial configuration is the same as the previous example, but you enter these commands:

```
serial interface 1
no frame-relay ip tcp header-compression
frame-relay map ip 131.108.177.178 178 nocompress
```

Use the **show frame-relay map** command display the resulting compression and encapsulation characteristics:

```
Router> show frame-relay map
Serial 1 (administratively down): ip 131.108.177.177 177
dlci 177(0xB1,0x2C10), static,
broadcast
CISCO
Serial 1 (administratively down): ip 131.108.177.178 178
dlci 178(0xB2,0x2C20), static
broadcast ISCO
```

The result of the commands is to disable header compression for th first map (with DLCI 177), which inherited its header compression characteristics from the interface, and also explicitly to disable header compression for the second map (with DLCI 178), which was explicitly configured for header compression.

Configure Discard Eligibility

You can specify which Frame Relay packets have low priority or low time sensitivity and will be the first to be dropped when a Frame Relay switch is congested. The mechanism that allows a Frame Relay switch to identify such packets is the discard eligibility (DE) bit.

This feature requires that the Frame Relay network be able to interpret the DE bit. Some networks take no action when the DE bit is set. Other networks use the DE bit to determine which packets to discard. The most desirable interpretation is to use the DE bit to determine which packets should be dropped first and also which packets have lower time sensitivity.

You can define DE lists that identify the characteristics of packets to be eligible for discarding, and you can also specify DE groups to identify the DLCI that is affected.

- In global configuration mode, define a DE list specifying which packets can be dropped when the Frame Relay switch is congested:

  frame-relay de-list ***list-number*** **{protocol** ***protocol*** **| interface** ***type number*}** ***characteristic***

You can specify DE lists based on the protocol or the interface, and on characteristics such as fragmentation of the packet, a specific TCP o User Datagram Protocol (UDP) port, an access list number, or a packet size.

- In interface configuration mode, define a DE group specifying th DE list and DLCI affected:

  frame-relay de-group ***group-number dlci***

Configure DLCI Priority Levels

DLCI priority levels allow you to separate different types of traffic and can provide a traffic management tool for congestion problems caused by these situations:

- Mixing batch and interactive traffic over the same DLC
- Traffic from sites with high-speed access being queued at destination sites with lower speed access

Before you configure the DLCI priority levels, complete these tasks:

1. Define a global priority list.
2. Enable Frame Relay encapsulation, as described earlier in this section.
3. Define static or dynamic address mapping, as described earlier in this section.
4. Make sure that you define each of the DLCIs to which you intend to apply levels. You can associate priority-level DLCIs with subinterfaces.
5. Configure the LMI, as described earlier in this section.

Note: ***DLCI priority levels provide a way to define multiple parallel DLCIs for different types of traffic. DLCI priority levels do not assign priority queues within the router or access server; in fact, they are independent of the device's priority queues. However, if you enable queuing and use the same DLCIs for queuing, then high-priority DLCIs can be put into high-priority queues.***

In interface configuration mode, configure DLCI priority levels by enabling multiple parallel DLCIs for different types of Frame Relay traffic, associating specified DLCIs with the same group, and defining their levels:

```
 frame-relay priority-dlci-group group-
number high-dlci medium-dlci normal-dlci
low-dlci
```

Note: ***If you do not explicitly specify a DLCI for each of the priority levels, the last DLCI specified in the command line is used as the value of the remaining arguments. However, you must provide at least the high-priority and the medium-priority DLCIs.***

Monitor the Frame Relay Connections

To monitor Frame Relay connections, perform any of these tasks in EXEC mode:

- Clear dynamically created Frame Relay maps, which are created by the use of Inverse ARP.

  ```
  clear frame-relay-inarp
  ```

- Display information about Frame Relay DLCIs and the LMI.

  ```
  show interfaces serial number
  ```

- Display LMI statistics.

  ```
  show frame-relay lmi [type number]
  ```

- Display the current Frame Relay map entries.

  ```
  show frame-relay map
  ```

- Display PVC statistics.

  ```
  show frame-relay pvc [type number [dlci]]
  ```

- Display configured static routes.

  ```
  show frame-relay route
  ```

- Display Frame Relay traffic statistics.

  ```
  show frame-relay traffic
  ```

- Display information about the status of LAPF.

  ```
  show frame-relay lapf
  ```

- Display all the SVCs under a specified map list.

  ```
  show frame-relay svc maplist
  ```

Example: Configuration Providing Backward Compatibility

This example configuration provides backward compatibility and interoperability with earlier versions that are not compliant with RFC 1490. The ietf keyword is used to generate RFC 1490 traffic. This configuration is possible because of the flexibility provided by separately defining each map entry.

```
encapsulation frame-relay
frame-relay map ip 131.108.123.2 48 broadcast ietf
```

Interoperability is provided by IETF encapsulation.

```
frame-relay map ip 131.108.123.3 49 broadcast ietf
frame-relay map ip 131.108.123.7 58 broadcast
```

This line allows the router to connect with a device running an older version of software.

```
frame-relay map decnet 21.7 49 broadcast
```

Configure IETF based on map entries and protocol for more flexibility. Use this method of configuration for backward compatibility and interoperability.

Example: Booting from a Network Server over Frame Relay

When booting from a Trivial File Transfer Protocol (TFTP) serve over Frame Relay, you cannot boot from a network server via a broadcast. You must boot from a specific TFTP host. Also, **`frame-relay map`** command must exist for the host that you will boot from.

For example, if file gs3-bfx is to be booted from a host with IP address 131.108.126.2, these commands would need to be in the configuration:

```
boot system gs3-bfx 131.108.126.2

interface Serial 0
encapsulation frame-relay
frame-relay map IP 131.108.126.2 100 broadcast
```

The **`frame-relay map`** command is used to map an IP address into a DLCI address. To boot over Frame Relay, you must explicitly give the address of the network server to boot from, and a frame-relay map entry must exist for that site. For example, if fil gs3-bfx.83-2.0 is to be booted from a host with IP address 131.108.126.111, these com mands must be in the configuration:

```
boot system gs3-bfx.83-2.0 131.108.13.111
!
interface Serial 1
ip address 131.108.126.200 255.255.255.0
encapsulation frame-relay
frame-relay map ip 131.108.126.111 100 broadcast
```

In this case, 100 is the DLCI that can get to host 131.108.126.111.

The remote router must have this **`frame-relay map`** entry:

```
frame-relay map ip 131.108.126.200 101 broadcast
```

This entry allows the remote router to return a boot image (from the network server) to the router booting over Frame Relay. Here, 101 is a DLCI of the router being booted.

Chapter 14: T1 Interface Configuration Steps

Introduction

This chapter describes how to configure the AI2524 for fractional T1.

Configure Fractional T1

Configuration Overview

This section describes how to configure fractional T1 and T1 (FT1/T1 service modules installed the AI2425 router. The tasks associated with configuring T1 include:

- Specify the Clock Source
- Enable Data Inversion Before Transmission
- Specify the Frame Type of a FT/T1 Line
- Specify the CSU Line Build Out
- Specify FT1/T1 Line-Code Type
- Enable Remote Alarms
- Enable Loopcodes that Initiate Remote Loopbacks
- Specify Timeslots

Specify the Clock Source

To specify the clock source for the FT1/T1 CSU/DSU internal clock or the line clock, type this in interface configuration mode:

```
service-module t1 clock source {internal |
line}
```

Enable Data Inversion Before Transmission

Data inversion is used to guarantee the T1s density requirement on an AMI line when using bit-oriented protocols such as High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), X.25, and Frame Relay.

To guarantee the ones density requirement on an AMI line using th FT1/T1 CSU/DSU module, type this in interface configuration mode:

```
service-module t1 data-coding inverted
```

This command inverts bit codes by changi ng all 1 bits to 0bits and all 0 bits to 1 bits.

If the timeslot speed is se t to 56kbps, this command is rejected because line density is guaranteed when transmitti ng at 56kbps. Use this command with the 64 kbps line speed. If you transmit inverted bit codes, both CSU/DSUs must have this command configured for successful communication.

To enable normal data transmission on a FT1/T1 network, type this in interface configuration mode:

```
service-module tx1 data-coding normal
```

or

```
no service-module t1 data-coding inverted
```

Specify the Frame Type of a FT/T1 Line

To specify the frame type for a line using the FT1/T1 CSU/DSU module, type this in interface configuration mode:

```
service-module t1 framing {sf | esf}
```

Note: ***Choose either D4 Super Frame (sf) or Extended Super Frame (esf).***

In most cases, the service provider determines which framing type, either **`esf`** or **`sf`**, is required for your circuit.

Specify the CSU Line Build Out

To decrease the outgoing signal strength to an optimum value for the telecommunication carrier network, type this in interface configuration mode:

```
service-module t1 lbo {-15 db | -7.5 db}
```

This command decreases the outgoing signal strength in decibels.

To transmit packets without decreasing outgoing signal strength, typ this in interface configuration mode:

```
service-module t1 lbo none
```

The ideal signal strength should be between -15 dB and -22 dB, which is calculated by adding the phone company loss, cable length loss, and line build out.

You may use this command in back-to-back configurations, but it is not needed on most actual T1 lines.

Specify FT1/T1 Line-Code Type

To configure the line code for the FT1/T1 CSU/DSU module, type this in interface configuration mode:

```
service-module t1 linecode {ami | b8zs}
```

Choose alternate mark inversion (AMI) or binary 8 zero substitution (B8ZS).

Configuring B8ZS is a method of ensuring the T1s density require ment on a T1 line by substituting intentional bipolar violations in bit positions four and seven for a sequence of eight zero bits. When the CSU/DSU is configured for AMI, you must guarantee the T1s density requirement in your router configuration using the **`service-module t1 data-coding inverted`** command or the **`service-module t1 timeslots speed 56`** command.

In most cases, your T1 service provider determines which line-cod type, either **`ami`** or **`b8zs`**, is required for your T1 circuit.

Enable Remote Alarms

To generate remote alarms (yellow alarms) at the local CSU/DSU or detect remote alarms sent from the remote CSU/DSU, type this in interface configuration mode:

```
service-module t1 remote-alarm-enable
```

Remote alarms are transmitted by the CSU/DSU when it detects an alarm condition, such as a red alarm (loss of signal) or blue alarm (unframed 1's). The receiving CSU/DSU then knows there is an error condition on the line.

With D4 super frame configured, a remote alarm condition is transmitted by setting the bit2 of each time slot to zero. For received user data that has the bit 2 of each time slot set to zero, the CSU/DSU will interpret the data as a remote alarm and interrupt data transmission, which explains why remote alarms are disabled by default. With Extended

Super Frame configured, the remote alarm condition is signalled out of band in the facility data link.

You can see if the FT1/T1 CSU/DSU is receiving a remote alarm (yellow alarm) by issuing the **`show service-module`** command.

To disable remote alarms, type this in interface configuration mode:

```
no service-module t1 remote-alarm-enable
```

Enable Loopcodes that Initiate Remote Loopbacks

To configure the remote loopback code used to transmit or accept CSU loopback requests, and type this in interface configuration mode:

```
service-module t1 remote-loopback full
```

To configure the loopback code used by the local CSU/DSU to generate or detect **`payload-loopback`** commands, type this in interface configuration mode:

```
service-module t1 remote-loopback payload
[alternate | v54]
```

Note: ***By entering the*** **`service-module t1 remote-loopback`** ***command without specifying any keywords, you enable the standard-loopup codes, which use a 1-in-5 pattern for loopup and a 1-in-3 pattern for loopdown.***

You can simultaneously configure the **`full`** and **`payload`** loopback points. However, only one loopback payload code can be configured at a time. For example, if you configure the **`service-module t1 remote-loopback payload alternate`** command, a payload v.54 request, which is the industry standard and default, cannot b transmitted or accepted. Full and payload loopbacks with standard-loopup codes are enabled by default.

The **`no`** form of this command disables loopback requests. For example, the **`no service-module t1 remote-loopback full`** command ignores all full-bandwidth loopback transmissions and requests. Configuring the no form of the command may not prevent telco line providers from looping your router in esf mode, because fractional T1/T1 telcos use facilities data-link messages to initiate loopbacks.

If you enable the **`service-module t1 remote-loopback`** command, the **`loopback remote`** commands on the FT1/T1 CSU/DSU module will not be successful.

Specify Timeslots

To define timeslots for FT1/T1 module, type this in interface configuration mode:

```
service-module t1 timeslots { range | all}
[speed {56 | 64}]
```

This command specifies which timeslots are used in fractional T1 operation and determines the amount of bandwidth available to the route in each timeslot.

The range specifies the DS0 timeslots that constitute the FT1/T1 channel. The range is from 1 to 24, where the first timeslot is numbered 1 and the last timeslot is numbered 24. Specify this field by using a series of subranges separated by commas. The timeslot range must match the timeslots assigned to the channel group. In most cases, the service provider defines the timeslots that comprise a channel group. Use th **`no`** form of this command to select all FT1/T1 timeslots transmitting at 64 kbps, which is the default.

To use the entire T1 line, enable the **`service-module T1 timeslots all`** command.

Chapter 15: 56/64-kbps Switched and Digital Data Services (DDS) Interface Configuration Steps

Introduction

This chapter describes how to configure 2- and 4-wire 56/64 kbps service modules. These tasks are described:

- Set the Clock Source
- Set the Network Line Speed
- Enable Scrambled Data Coding
- Change Between DDS and Switched Dial-Up Modes
- Enable Acceptance of a Remote Loopback Request
- Select a Service Provider

Set the Clock Source

In most applications, the CSU/DSU should be configured with the **`service-module 56k clock source line`** command. Fo back-to-back configurations, use the **`internal`** keyword to configure one CSU/DSU and use the **`line`** keyword to configure the othe CSU/DSU.

Configure the clock source for a 4-wire 56/64-kbps CSU/DSU module:

```
service-module 56k clock source {line |
internal}
```

Do not use any form of this command to revert to the default clock source, which is the line clock.

Set the Network Line Speed

In interface configuration mode, configure the network line speed for a 4-wire 56/64-kbps CSU/DSU module:

```
service-module 56k clock rate line-speed
```

You can use the following line speed settings: 2.4, 4.8, 9.6, 19.2, 38.4, 56, 64 kpbs, and an auto setting.

The 64-kbps line speed cannot be used with back-to-back digital data service (DDS) lines. The subrate line speeds are determined by the service provider

Only the 56-kbps line speed is available in switched mode. Switched mode is the default on the 2-wire CSU/DSU and is enabled by the **`service-module 56k network-type`** interface configuration command on the 4-wire CSU/DSU.

The auto linespeed setting enables the CSU/DSU to decipher current line speed from the sealing current running on the network. Becaus back-to-back DDS lines do not have sealing current, use the auto setting only when transmitting over telco DDS lines and using the line clock as the clock source.

Do not use any form of this command to enable a network line speed of 56 kbps, which is the default.

Warning: If the console line speed is changed and saved to NVRAM and the router is reloaded, the router displays this message `Failed to change line0' speed.` ***The ne line speed is stored in the start-up configuration but not in the running configuration. In addition, changing the line speed on the router prevents connections to the console interface via the AI185DP (the line speed on the router and the AI185 must be set to 9600 bps). The default consol speed for the Cisco router is also set at 9600 bps.***

Enable Scrambled Data Coding

In interface configuration mode, prevent application data from repli cating loopback codes when operating at 64-kbps on a 4-wire CSU/DSU by scrambling bit codes before transmission:

```
service-module 56k data-coding scrambled
```

Enable the scrambled configuration only in 64-kbps digital data service (DDS) mode. If the network type is set to switched, the configuration is refused.

If you transmit scrambled bit codes, both CSU/DSUs must have this command configured for successful communication.

In interface configuration mode, enable normal data transmission fo the 4-wire 56/64-kbps module, which is the default:

```
service-module 56k data-coding normal
```

or

```
no service-module 56k data-coding
```

Change between DDS and Switched Dial-Up Modes

In interface configuration mode, transmit packets in switched dial-up mode or DDS mode using the 4-wire 56/64-kbps CSU/DSU module:

```
service-module 56k network-type dds
```

or

```
service-module 56k network-type switched
```

Do not use any form of these commands to transmit from a dedicated leased line in DDS mode. DDS is enabled by default for the 4-wir CSU/DSU. Switched mode is enabled by default for the 2-wire CSU/DSU.

In switched mode, you need additional dialer configuration commands to configure dial-out numbers. Before you enable the **`service-module 56k network-type switched`** command, both CSU/DSUs must use a clock source coming from the line and the clock rate must be configured to auto or 56k kbps. If the clock rate is not set correctly, this command will not be accepted.

The 2-wire and 4-wire 56/64-kbps CSU/DSU modules use V.25 bis dial commands to interface with the router. Therefore, the interfac must be configured using the **`dialer in-band`** command. DTR dial is not supported.

Warning:* *Any loopbacks in progress are terminated when switching between modes.

Enable Acceptance of a Remote Loopback Request

In interface configuration mode, enable the acceptance of a remot loopback request on a 2- or 4-wire 56/64-kbps CSU/DSU module:

```
service-module 56k remote-loopback
```

The **no service-module 56k remote-loopback** command prevents the local CSU/DSU from being placed into loopback by remote devices on the line. Unlike the T1 module, the 2- or 4-wire 56/64-kbps CSU/DSU module can still initiate remote loopbacks with the no form of this command.

Select a Service Provider

In interface configuration mode, select a service provider to use with a 2- or 4-wire 56/64 kbps dial-up line:

```
service-module 56k switched-carrier {att |
other | sprint}
```

The **att** keyword specifies AT&T or another digital network service provider as the line carrier, which is the default for the 4-wire 56/64 kbps CSU/DSU module. Th **sprint** keyword specifies Sprint or another service provider whose network carries mixed voice and data as the line carrier, which is the default for the 2-wire switched 56-kbps CSU/DSU module.

In a Sprint network, echo-canceler tones are sent during call setup to prevent echo cancelers from damaging digital data. The transmission of these cancelers may increase call setup times by 8 seconds on the 4-wire module. Having echo cancellation enabled does not affect dat traffic.

This configuration command is ignored if the network type is DDS.

Use the **no** form of this command to enable the default service provider. AT&T is enabled by default on the 4-wire 56/64 module. Sprint is enabled by default on the 2-wire switched 56 module.

Chapter 16: Basic Configuration

Connecting to the Network

Connecting to an Ethernet Network

The AI2524 can be connected to an Ethernet network by:

- Using a straight-through 10BaseT cable to connect the 10BaseT port to a 10BaseT hub.
- Using a crossover 10BaseT cable to connect the 10BaseT port to a PC network interface card.

Connecting to a WAN

- If you have a 4-wire 56K/64K DSU/CSU module, use a straight-through RJ-48S-to-RJ48S cable to connect the RJ-48S port to an RJ48S jack.
- If you have a FT1/T1 DSU/CSU module, use a straight-through RJ-48C-to-RJ48C cable to connect the RJ-48C port to an RJ48C jack.
- If you have a synchronous serial module, use a transition cable to connect the synchronous serial port to a modem or DSU/CSU.

Configuring

This chapter describes how to configure the AI2524 router and describes the following:

- Booting the Router for the First Time
- Configuring the Router
- Specifying the Boot Method
- Checking the Configuration Settings

This chapter provides just enough information to get the router up and running. Review the previous detailed configuration chapters for more information

Booting the Router for the First Time

Each time you power on the router, it goes through this boot sequence:

1. The router goes through power-on self-test diagnostics to verify basic operation of the CPU, memory, and interfaces.

2. The system bootstrap software (boot image) executes and searches for a valid Cisco IOS image (router operating system software). The source of the Cisco IOS image (Flash memory or a Trivial File Transfer Protocol [TFTP] server) is determined by the configuration register setting. The factory-default setting for the configura tion register is 0x2102, which indicates that the router should attempt to load a Cisco IOS image from Flash memory.

3. If after five attempts a valid Cisco IOS image is not found in Flash memory, the router reverts to boot ROM mode (which is used to install or upgrade a Cisco IOS image).

4. If a valid Cisco IOS image is found, then the router searches for a valid configuration file.

5. If a valid configuration file is not found in NVRAM, the router runs the System Configuration Dialog so you can configure it manually. For normal router operation, there must be a valid Cisco IOS image in Flash memory and a configuration file in NVRAM.

The first time you boot your router, you will need to configure the router interfaces and then save the configuration to a file in NVRAM. Proceed to the next section, Configuring the Router, for configuration instructions.

Configuring the Router

You can configure the router using one of these procedures:

Configuration mode recommended if you are familiar with Cisco IOS commands.

- AutoInstall-Recommended for automatic installation if anothe router running Cisco IOS software is installed on the network. This configuration method must be set up by someone with experience using Cisco IOS software.

- System Configuration Dialog-Recommended if you are not familiar with Cisco IOS commands.

- Use the procedure that best fits the needs of your network configuration and level of Cisco IOS experience.

Warning: Acquire the correct network addresses from your system administrator or consult your network plan to determin correct addresses before you begin to configure the router.

Using Configuration Mode

You can configure the router manually if you prefer not to use AutoInstall or the System Configuration Dialog. Take these steps to configure the router manually:

1. Connect a console terminal following the instructions in the section Connecting the Console Terminal and Modem in the chapter Installing the Cisco 2524 Router, and then power ON the router.
2. When you are prompted to enter the initial dialog, enter no to go into the normal operating mode of the router:

```
Would you like to enter the initial dialog? [yes]: no
```

3. After a few seconds you will see the user EXEC prompt (Router). Enter the **`enable`** command to enter enable mode. You can only make configuration changes in enable mode.

```
Router> enable
The prompt changes to the privileged EXEC (enable) prompt:
Router#
```

4. Enter the **`configure terminal`** command at the enable prompt to enter configuration mode:

```
Router# configure terminal
You can now enter any changes you want to the configuration.
```

5. Press <Ctrl-Z-Z> to exit configuration mode.

To see the current operating configuration, enter the **`show running-config`** command at the enable prompt:

```
Router# show running-config
```

6. To see the configuration in NVRAM, enter th **show startup-config** command at the enable prompt:

```
Router# show startup-config
```

The results of the **show running-config** and **show startup-config** commands will be different if you have mad changes to the configuration but have not yet written them to NVRAM.

7. To make your changes permanent, enter the **copy running-config** startup-config command at the enable prompt:

```
Router# copy running-config startup-config
*******
```

The router is now configured and will boot with the configuration you entered.

Using AutoInstall

The AutoInstall process is designed to configure the router automatically after connection to your WAN. For AutoInstall to work properly, a Transmission Control Protocol Internet Protocol (TCP/IP) host on your network must be reconfigured to provide the required configuration files. The TCP/IP host may exist anywhere on the network as long as these conditions are maintained:

1. The host must be on the remote side of the router's synchronous serial connection to the WAN.
2. User Datagram Protocol (UDP) broadcasts to and from the router and the TCP/IP host must be enabled.

This functionality is coordinated by your system administrator at th site where the TCP/IP host is located. You should not attempt to us AutoInstall unless the required files have been provided on the TCP/IP host.

AutoInstall works on synchronous serial connections only.

Take these steps to prepare your router for the AutoInstall process:

3. Attach the WAN cable to the router.

4. Turn ON power to the router.

   The router will load the operating system image from Flash memory. If the remote end of the WAN connection is connected and properly configured, the AutoInstall process will begin.

   If AutoInstall successfully completes, you can write the configuration data to the router's NVRAM.Perform this step to complete this task.

5. Enter the **`copy running-config startup-config`** command:

```
Router# copy running-config startup-config
```

Taking this step saves the configuration settings that the AutoInstall process created in the router. If you do not do this, your configuration will be lost the next time you reload the router.

Using the System Configuration Dialog

If you do not plan to use AutoInstall, make sure all the WAN cables are disconnected from the router. This will prevent the router from attempting to the run the AutoInstall process. The router will attempt to run AutoInstall whenever you power it on if there is a WAN connection on both ends and the router does not have a configuration fil stored in NVRAM. It can take several minutes for the router to determine that AutoInstall is not set up to a remote TCP/IP host.

If your router does not have a configuration (setup) file and you are not using AutoInstall, the router will automatically start the setup com mand facility. An interactive dialog called the System Configuration Dialog appears on the console screen. This dialog helps you navigat through the configuration process by prompting you for the configuration information necessary for the router to operate.

Many prompts in the System Configuration Dialog include default answers, which are included in square brackets following the question.

To accept the default answer, press <Enter>; otherwise enter your response.

This section gives an example configuration using the System Configuration Dialog. When you are configuring your router, respond as ap propriate for your network.

At any time during the System Configuration Dialog, you can request help by typing a question mark (?) at a prompt.

Before proceeding with the System Configuration Dialog, obtain from your system administrator the node addresses and the number of bits in the subnet field (if applicable) of the Ethernet and synchronous serial ports.

Take these steps to configure the router using the System Configura tion Dialog:

1. Connect a console terminal to the console connector on the rear panel of your router, and turn ON power to the router. (For mor information, refer to the section Connecting the Console Terminal and Modem in the chapter Installing the Cisco 2524 Routers.)

   The default parameters for the console port are 9600 baud, 8 data bits, no parity, and 2 stop bits.

2. After about 30 seconds, information similar to this is displayed on the console screen.

   The messages displayed vary, depending on the Cisco IOS releas and feature set you selected. The screen displays in this section are for reference only and may not exactly reflect the screen displays on your console.

When you see this information, you have successfully booted your router:

```
System Bootstrap, Version X.X(XXXX) [XXXXX XX], RELEASE SOFTWARE
Copyright (c) 1986-1992 by Cisco Systems
2500 processor with 4096 Kbytes of main memory
Notice: NVRAM invalid, possibly due to write erase
F3: 5797928+162396+258800 at 0x3000060
                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 DEARS. 252.227-7013.

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

Cisco Internetwork Operating System Software
IOS (to) X000 Software (IGS-J-L), Version XX.X(XXXX) [XXXXX XXX]
Copyright (c) 1986-1996 by Cisco Systems, Inc.
Compiled Fri 20-Oct-95 16:02 by XXXXX
Image text-base: 0x03030FC0, data-base: 0x00001000
Cisco 252X (68030) processor (revision A) with 4092K/2048K bytes of memory.
Processor board ID 00000000
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version X.X, NET2, 8FE and GOSIP compliant.
TN3270 Emulation software (copyright 1994 by TGV Inc).
Basic Rate ISDN software, Version X.X.
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
1 ISDN Basic Rate interface.
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read ONLY)

Notice: NVRAM invalid, possibly due to write erase.
         --- System Configuration Dialog

At any point you may enter a question mark '?' for help.
Refer to the 'Getting Started' Guide for additional help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
Would you like to enter the initial configuration dialog? [yes]:
```

3. Press <Enter> or enter yes to begin the configuration process.

4. When the System Configuration Dialog asks whether you want to view the current interface summary, press< Return> or enter yes:

```
First, would you like to see the current interface summary? [yes]:
Any interface listed with OK? value “NO" does not have a valid configuration
Interface IP-Address OK? Method  Status    Protocol
EthernetO unassigned  NO not set    up     down
BRIO      unassigned  NO not set    up     up
SerialO   unassigned  NO not set  down     down
Seriall   unassigned  NO not set  down     down
```

5. Configure the global parameters. Choose which protocols to sup port on the Ethernet interface. For IP installations, you can press <Enter> to accept the default values (in brackets) for most of the questions. A typical configuration is:

```
Configuring global parameters:
Enter host name {Router
```

Next, you are prompted to enter an enable secret password. There are two types of privileged-level passwords:

Enable secret password (a very secure, encrypted password

Enable password (a less secure, nonencrypted password)

The enable password is used when the enable secret password does not exist.

For maximum security, be sure the passwords are different. If you enter the same password for both, the router will accept your entry, but will display a warning message indicating that you should enter a different password.

6. Enter an enable secret password:

   The enable secret is a one-way cryptographic secret used instead of the enable password when it exists.

```
 Enter enable secret: pail
```

The enable password is user when there is no enable secret and when using older software and some boot images.

7. Enter the enable and virtual terminal passwords:

```
Enter enable password:  shovel
Enter virtual terminal password: vterml
```

8. Press <Enter> to accept Simple Network Management Protocol (SNMP) management, or enter no to refuse it:

```
Configure SNMP Network Management? [yes]:  no
```

9. In this example, the router is configured for AppleTalk, IP, and IPX. Configure the appropriate protocols for your router:

```
Configure Vines? [no]:
Configure LAT? [no]:
Configure AppleTalk? [no]: yes
    Multizone networks? [no]: yes
Configure DECnet? [no]:
Configure IP? [yes]:
   Configure IGRP routing? [yes]:
   Your IGRP autonomous system number [l]: 15
Configure CLNS? [No]:
Configure bridging? [no]:
Configure IPX? [no]: yes
Configure XNS? [no]:
Configure Apollo? [no]:
```

10. Enter the ISDN BRI switch type for the router. The ISDN switch type appropriate for the router depends on the ISDN provider's equipment. This table lists the ISDN switch types:

```
Enter ISDN BRI Switch Type [none]: besic-5ess
```

Country	ISDN Switch Type	Description
Australia	basic-ts013	Australian TS013 switches
Europe	basic-1tr6	German 1TR6 ISDN switches
	basic-nwnet3	Norwegian NET3 ISDN switches (phase 1)
	basic-net3	NET3 ISDN switches (UK and others)
	vn2	French VN2 ISDN switches

Country	ISDN Switch Type	Description
	vn3	French VN3 ISDN switches
Japan	ntt	Japanese NTT ISDN switches
North America	basic-5ess	AT&T basic rate switches
	basic-dms100	NT DMS-100 basic rate switches
	basic-ni1	National ISDN-1 switches
New Zealand	basic-nznet3	New Zealand NET3 switch

Configuring the Ethernet or Token Ring Interfaces

Take these steps to configure the Ethernet or Token Ring interface to allow communication over a LAN. To configure the interface param eters, you need to know your Ethernet or Token Ring interface network addresses. In this example, the system is being configured for an Ethernet LAN using IP.

1. Respond (using your addresses and subnet mask) to the setup prompts, substituting the correct addresses and host names as appropriate:

```
Configuring interface Ethernet0:
Is this interface in use? [yes]:
Configure IP on this interface? [yes]:
IP address for this interface: 172.16.72.1
Number of bits in subset field [8]: 8
Class s network is 172.16.0.0, 8 subnet bits; mask i
     255.255.255.0
```

2. Enter yes if you will be using AppleTalk on the interface. Enter yes to configure the router for extended AppleTalk networks, and then enter the cable range. Enter the zone name, and any other additional zones that will be associated with your local zone:

```
Configure AppleTalk on this interface? [no]: yes
Extended AppleTalk network? [no]: yes
AppleTalk starting cable range [0]: 3
AppleTalk ending cable range [1]: 3
AppleTalk zone name [myzone]:
AppleTalk additional zone name: otherzone
AppleTalk additional zone name:
```

3. Determine if you are going to enable IPX on the interface. If so, enter yes and then enter the unique IPX network number:

```
Configure IPX on this interface? [no]: yes
IPX network number [1]: B001
```

Configuring the Synchronous Serial Interfaces

The synchronous serial interfaces are configured to allow connection to WANs. Once the Ethernet or Token Ring port on your router has been configured, take these steps to configure the synchronous serial interfaces:

1. Press <Enter> or enter yes to configure serial port 0:

```
Configuring interface Serial0:
   Is this interface in use? [yes]:
```

2. Determine which protocols you want on the synchronous serial interface and enter the appropriate responses. In this example, th system is being configured for IP, AppleTalk, and IPX:

```
Configure IP on this interface? [yes:
Configure IP unnumbered on this interface? Loo]:
   IP address for this interface: 172.16.73.1
   Number of bits in sunned field [8]:
   Class B network is 172.16.0.0, 8 subnet bits; mask is
    255.255.255.0
Configure AppleTalk on this interface? [no]: yes
    Extended AppleTalk network? [yes]:
    AppleTalk starting cable range [2]: 4
    AppleTalk ending cable range [3]: 4
    AppleTalk zone name [myzone]: ZZ Serial
    AppleTalk additional zone name:
Configure IPX on this interface? [no]: yes
     IPX network number [2]: B002
```

3. Configure the second synchronous serial interface, for example, as:

```
Configuring interface Seriall:
   Is this interface in use? [yes]:
   Configure IP on this interface? ~yes;:
   Configure IP unnumbered on this interface? [no]:
     IP address for this interface: 172.16.74.2
     Number of bits in subset field [8]:
     Class B network is 172.16.0.0, 8 subset bits; mask is
     255.255.255.0
Configure AppleTalk on this interface? [no]: yes
    AppleTalk starting cable range [3]: 5
    AppleTalk ending cable range [4]: 5
    AppleTalk zone name [myzone]: ZZ Serial
    AppleTalk additional zone name:
Configure IPX on this interface? [no]: yes
    IPX network number [3]: B003
```

4. The configuration you enter is now displayed and you are asked if you want to use the displayed configuration. If you enter no, you will lose the configuration information you just entered and you can begin the configuration again. If you enter yes, the configuration will be entered and saved in the startup configuration:

```
Use this configuration? [yes/no]: yes
Building configuration...
Use the enabled mode 'configure' command to modify this configuration.

    Press RETURN to get started!
```

Configuring ISDN

If you have an ISDN BRI WAN module, configure the BRI port for ISDN. This section explains typical ISDN configurations for one or two B channels. In the examples, the BRI port is configured for IP routing and Point-to-Point Protocol (PPP) encapsulation.

Complete these steps to configure the router for a basic ISDN PPP connection on a single B channel or two B channels, substituting th correct address and host names as appropriate for your network.

1. Enter enable mode:

```
Router> enable
Password: enable password
```

2. Enter the **`configure terminal`** command.

```
Router# config term
```

3. If you have not already done so, enter the **`isdn switch-type`** command to configure the ISDN switch type:

```
router (config) # ISDN switch-type switch-type
```

4. Enter the BRI interface, encapsulation method (PPP), authentication type, target router's IP address and ISDN number to dial, and the dialer group number:

```
Router (config) # interface bri 0
Router (config-if) # encapsulation ppp
Router (config-if) #ppp authentication chap
Router (config-if) #  dial map ip targetrouter_ipaddress targetrouter_phonenumber
Router (config-if) # dialer-group groupnumber
```

Do not use periods or hyphens when you are entering dialing numbers.

Note: ***The ISDN/BRI interface provides dial backup for the AI2524 card. When a connection is requested, the system checks the username presented for validity, then dials back the number associated with the username.***

5. Some ISDN switch types, such as Basic NI1 or DMS-100 switch service, require you to configure a service profile identifie (SPID). Enter the SPID information substituting the appropriat entries for your installation:

```
Router (config-if) # isdn spid1 SPID_no phone_number
Router (config-if) # isdn spid2 SPID_no phone_number
```

6. To set up a second B channel for bandwidth on demand, enter the load-threshold command to set the ISDN load threshold. The load threshold determines the percentage of network loading at which the second ISDN B channel is triggered. The value ranges from 1 to 255 (100 percent).

```
Router (config-if) 3 dialer load-threshold 128
```

In this example, the value of 128 means that when the first B channel reaches 50 percent of its bandwidth capacity (128 equals 50 percent of 255), the second B channel will be activated to assist with the bandwidth load.

7. Enter the **`access-list`** command to configure the ISDN line to come up whenever IP packets are to be sent:

```
Router (config-if) # access-list access-list-number  permit-ip sourcerouter-
ipnetwork sourcerouter-subnetmask targetrouter-ipnetwork targetrouter-subnetmask
Router (config) # dialer-list groupnumber list access-list-number
```

8. Configure a static route to allow connectivity to the target router's local network. Enter the network number of the target router's local IP network and subnet mask, and the IP address of the target router's BRI port:

```
Router (config) 3 ip route targetrouter_ipnetwork subnetmask
targetBRIport_ipaddress
```

9. Enter the **`exit`** command to exit configuration mode.

10. Enter the **`copy running-config startup-config`** command to save the configuration to NVRAM.

Configuring Switched 56

This section explains how to configure the 4-wire 56/64-kbps DSU/CSU WAN modules for switched 56-kbps circuit-switched service The 4-wire 56/64-kbps DSU/CSU WAN module is configured for DDS as the factory default, but it can be configured for either switched 56/64-kbps service or DDS.

Take these steps to configure the 4-wire 56/64-kbps DSU/CSU WAN module for circuit-switched service, substituting the correct addresses and host names as appropriate for your network:

1. Enter enable mode:

```
Router> enable
password: enablepassword
```

2. Enter configuration mode:

```
Router# config term
Router(config)#
```

3. Assign an IP address to the serial port on the module:

```
Router(config)# interface serial port_number
Router(config-if)# ip address ipaddress suLnetmask
Router(config-if)# no keepalive
```

4. Set the network type to switched:

```
Router(config-if)# service-module 56k network-type switched
```

5. Set the carrier type, where carrier can be ttt, sprint, or other:

```
Router(config-if)# service-module 56k switched-carrier carrier
```

6. Enter the dialer information:

```
Router(config-if)# dialer in-band
Router(config-if)# dialer string targetrouter_phonenumber
Router(config-if)# dialer-group~ groupnumber
Router(config-if)# exit
Router(config)# dialer-list groupnumber protocol protocol permit
Router(config)#
```

7. Return to user EXEC mode:

```
Router(contlg)# exit
Router# exit
Router>
```

Configuring DDS

The 4-wire 56/64-kbps DSU/CSU WAN module is configured for DDS (which are leased or dedicated lines) as the factory default, but it can be configured for either switched 56/64-kbps service or DDS. Th DDS configuration is described in this section. To configure the 4-wire 56/64-kbps DSU/CSU WAN module for circuit-switched service, follow the instructions in the previous section Configuring Switched 56.

Take these steps to configure the 4-wire 56/64-kbps DSU/CSU module for DDS, substituting the correct addresses and host names as appropriate for your network

1. Enter enable mode:

```
Router> enable
password: enablepassword
```

2. Enter configuration mode:

```
Router# config term
Router(config)#
```

3. Assign an IP address to the serial port on the module:

```
Router(config)# interface serial port _number
Router(config-if)# ip address ipaddress subnet mask
Router(config-if)# no keepalive
```

4. Set the network type to DDS:

```
Router(config-if)# service-module 56k network-type ads
```

5. Return to user EXEC mode:

```
Router(config-if)# exit
Router(config)# exit
Router# exit
Router>
```

Configuring the Fractional T1/T1 DSU/CSU WAN Module

This section describes how to configure the fractional Tl/T1 DSU/CSU WAN module is configured for Extended Superframe Format (ESF) signal format, bipolar zero substitution (B8ZS), and full bandwidth as the factory default. Depending on networking environment, you might need to change these settings.

Take these steps to configure the fractional Tl/T1 DSU/CSU WAN module for a typical leased-line connection, substituting the correct addresses and host names as appropriate for your network:

1. Enter enable mode:

```
Router> enable
password: enablepassword
```

2. Enter configuration mode:

```
Router# config term
Router(config)#
```

3. Assign an IP address to the serial port on the module:

```
Router(config)# interface serial port_number
Router(config-if)# ip address ipaddress suLnetmask
Router(config-if)# no keepalive
```

4. Enter the framing type and line code type, substituting framing_type with **sf** (Superframe) o **esf** (Extended Super frame) and linecode_type with **ami** (alternate mark inversion) o **b8zs** (bipolar eight zero substitution):

```
Router(config-if)# service-module tl framing framing_type
Router(config-if)# service-module tl linecode linecode type
```

5. If you are using fractional T1 service, enter the time slot range and speed. In this example, the time slot range is from 1 to 20 and the speed is 64-kbps:

```
Router(config.f)# service-module tl timeslots 1-20 speed 64
```

6. Return to user EXEC mode:

```
Router(config-if)# exit
Router(config)# exit
Router# exit
Router>
```

Specifying the Boot Method

You can enter multiple boot commands in the configuration in NVRAM to provide a backup method for loading the Cisco IOS image onto the router. The router boots using the first boot command that succeeds. If you enter multiple boot commands, the router executes them in the order they are entered. There are two ways to load the Cisco IOS image: from Flash memory or from a TFTP server on the network.

1. Flash memory

   Information stored in Flash memory is not vulnerable to network failures that might occur when you load system software from servers. In this example, replace filename with the filename of the Cisco IOS image:

```
Router> enable
Password: enablepassword
Router# configure terminal
Router (config)# boot system flash filename
Router (config)# Ctrl-Z
Router# copy running-config startup-config
Building configuration...
 [OK]
Router# exit
Router>
```

2. TFTP server

   If Flash memory is not available, or if Flash memory does not contain a valid Cisco IOS image, you can specify that system software be loaded from a TFTP server on your network as a backup boot method for the router. In this example, replace filename with the filename of the Cisco IOS image, and replace IP address with the IP address of the TFTP server:

```
Router> enable
Password: enablepassword
Router# configure terminal
Router (config)# boot system tftp filename ipaddress
Router (config)# Ctrl-Z
Router# copy running -config startup-config
Building configuration ...
 [OK]
Router# exit
Router>
```

Checking the Configuration Settings

Enter the **`show version`** command to check the software version (third line from the top in this display) and configuration register setting (at the end of this display):

```
Router> shovrsion
 Cisco Internetwork Operating System Software
 IOS (to) XX00 Software (XXX-X-X), RELEASE SOFTWARE XX.X(XXXX) [XXX]
 Copyright (c) 1986-1996 by Cisco Systems, Inc.
 Compiled Tue XX-XXX-XX 13:07 by XXXXX
 Image text-base: 0x03032810, data-base: 0x00001000

  ROM: System Bootstrap, Version X.X(XXXX) [XXXXX], RELEASE SOFTWARE
  ROM: XX00 Bootstrap Software (XXX-BOOT-X), Version XX.X(XXXXX) [XXXXX]

  Router uptime is 4 minutes
  System restarted by power-on
  System image file is Rflash:XXX/XXX-X-X.Novl4", booted via flash

  Cisco XXXX(68030) processor (revision X) with 4092K/2048K bytes of memory.
  Processor board ID 00000000
  Bridging software.
  SuperLAT software copyright l99X by Meridian Technology Cord).
  X.25 software, Version X.X, NET2, BEE and GOSIP compliant.
  TN3270 Emulation software (copyright l99X by TGV Inc).
  1 Ethernet/IEEE 802.3 interface.
  2 Serial network interfaces.
  No module installed for Serial Interface 0
  No module installed for Serial Interface 1
  32K bytes of non-volatile configuration memory.
  8192K bytes of processor board System flash (Read ONLY)

  Configuration register is 0x2102

  Router>
```

Chapter 17: Command References

Introduction

This chapter contains a link to the AI2524/Cisco IOS v. 11.2 documentation.

Command References

Refer to the Cisco Command Reference chapters on the AI2524/Cisco IOS v. 11.2 documentation CD.

- Security Command Referenc
- Wide-Area Networking Command Reference
- Network Protocols Command Reference Part 1
- Network Protocols Command Reference Part 2
- Network Protocols Command Reference Part 3
- Bridging and IBM Networking Command Reference

Chapter 18: System Error Messages

Introduction

This chapter contains a link to the AI2524/Cisco IOS v. 11.2 documentation.

System Error Messages

Refer to the Cisco Command Reference chapters on the AI2524/Cisco IOS v. 11.2 documentation CD.

Chapter 19: Debug Command Reference

Introduction

This chapter contains a link to the AI2524/Cisco IOS v. 11.2 documentation.

Debug Command Reference

Refer to the Cisco Command Reference chapters on the AI2524/Cisco IOS v. 11.2 documentation CD.

Appendix A: Release Notes

AISwitch Release Notes

AI2524, Version 1.00

Router Card

August 1997

Applied Innovation, Inc.
5800 Innovation Drive
Dublin, Ohio 43016-3271
(614) 798-2000
(800) 247-9482
FAX (614) 798-1770

Copyright

FCC Warning

The Federal Communications Commission has set limits for emitted radio interference, and the AISwitch is constructed with this electromagnetic interference (EMI) limitation in mind. The AISwitch is classified under FCC regulations as a Class A device, that is, a device for use in commercial environments and not in residential areas. This device has been tested and shown to comply with the following FCC rule: Part 15 Subpart J. Operation of this equipment in a residential area may cause interference to radio and TV reception, requiring the user to take whatever steps are necessary to correct the interference.

Information is available from the FCC describing possible corrective actions. To maintain low EMI levels, we suggest that you use only metal connectors and shielded cable grounded to the frame.

Electrostatic Discharge Warning

The AISwitch and its peripherals contain electrostatic sensitive components. Proper handling, shipping, and storage precautions must be exercised:

- *Removal and installation of circuit boards must be performed in a static-free environment. This means the technician should wear an anti-static wrist strip and stand on an anti-static mat. Both the wrist strap and mat must be grounded at the same point as th AISwitch enclosure.*
- *When not in use, circuit boards must be kept in their anti-static plastic bags.*
- *Circuit boards must only be removed from their anti-static plastic bags immediately prior to installation into the AISwitch enclosure.*
- *Immediately upon removal from the enclosure, circuit boards must be inserted into their anti-static bags.*
- *Do not ship or store the electronic circuit boards near strong electrostatic, electromagnetic, magnetic, or radioactive fields.*

Applied Innovation, Inc.
5800 Innovation Drive
Dublin, Ohio 43016-3271
(614) 798-2000
(800) 247-9482
FAX (614) 798-1770

AISwitch Release Notes

AI2524, Version 1.00

Router Card

August 1997

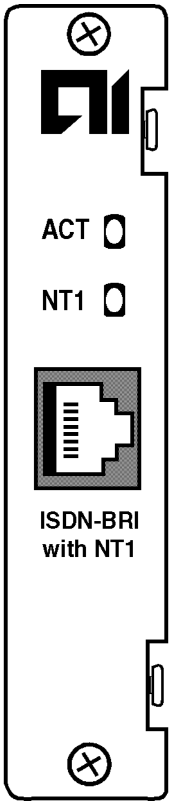
New Features

These release notes document new or enhanced features and com mands, upgrade instructions, and problem resolutions for the AI2524 Router card.

- ISDN/BRI Inferface

Instructions for use of this product are detailed in the AI2524 Router Manual, document number 2524UM.

ISDN/BRI Inferface



The AI2524 ISDN/BRI module provides the digitalization of telephone network so that voice, data, text, graphics, music, video, and other source material can be provided to end users from a single end-user terminal over existing telephone wiring.

ACT	Activity LED Flashing LED indicates normal opera tion. Transmitting and receiving dat normally.
NT1	Network Termination 1 LED If the NT1 LED is on, the router detects the ISDN link integrity signal at the U interface and the internal S/T interface, indicating that an ISDN connection has been established. If the NT1 LED blinks once per second, the ISDN connection at the U interface is up and the internal S/T interface is coming up. If this condition persists, th ISDN port is either not configured o configured incorrectly. If the NT1 LED blinks 8 times per second, the ISDN connection at the internal S/T interface is up and the U interface is coming up. If the NT1 LED is off, the router is not detecting the ISDN link integrity signal. Check the BRI cable connection.
ISDN-BRI with NT1	RJ-45 connector

Configuring ISDN

If you have an ISDN BRI WAN module, configure the BRI port for ISDN. This section explains typical ISDN configurations for one or two B channels. In the examples, the BRI port is configured for IP routing and Point-to-Point Protocol (PPP) encapsulation.

Complete these steps to configure the router for a basic ISDN PPP connection on a single B channel or two B channels, substituting th correct address and host names as appropriate for your network.

1. Enter enable mode:

```
Router> enable
Password: enable password
```

2. Enter the **`configure terminal`** command.

```
Router# config term
```

3. If you have not already done so, enter the **`isdn switch-type`** command to configure the ISDN switch type:

```
router (config) # ISDN switth-type switch-type
```

4. Enter the BRI interface, encapsulation method (PPP), authentication type, target router's IP address and ISDN number to dial, and the dialer group number:

```
Router (config) # interface bri 0
Router (config-if) # encapsulation ppp
Router (config-if) #ppp authentication chap
Router (config-if) #  dial map ip targetrouter_ipaddress targetrouter_phonenumber
Router (config-if) # dialer-group groupnumber
```

Do not use periods or hyphens when you are entering dialing numbers.

Note: ***The ISDN/BRI interface provides dial backup for the AI2524 card. When a connection is requested, the system checks the username presented for validity, then dials back the number associated with the username.***

5. Some ISDN switch types, such as Basic NI1 or DMS-100 switch service, require you to configure a service profile identifie (SPID). Enter the SPID information substituting the appropriat entries for your installation:

```
Router (config-if) # isdn spid1 SPID_no phone_number
Router (config-if) # isdn spid2 SPID_no phone_number
```

6. To set up a second B channel for bandwidth on demand, enter the load-threshold command to set the ISDN load threshold. The load threshold determines the percentage of network loading at which the second ISDN B channel is triggered. The value ranges from 1 to 255 (100 percent).

```
Router (config-if) 3 dialer load-threshold 128
```

In this example, the value of 128 means that when the first B channel reaches 50 percent of its bandwidth capacity (128 equals 50 percent of 255), the second B channel will be activated to assist with the bandwidth load.

7. Enter the **`access-list`** command to configure the ISDN line to come up whenever IP packets are to be sent:

```
Router (config-if) # access-list access-list-number  permit-ip sourcerouter-
ipnetwork sourcerouter-subnetmask targetrouter-ipnetwork targetrouter-subnetmask
Router (config) # dialer-list groupnumber list access-list-number
```

8. Configure a static route to allow connectivity to the target router’s local network. Enter the network number of the target router’s local IP network and subnet mask, and the IP address of the target router’s BRI port:

```
Router (config) 3 ip route targetrouter_ipnetwork subnetmask
targetBRIport_ipaddress
```

9. Enter the **`exit`** command to exit configuration mode.

10. Enter the **`copy running-config startup-config`** command to save the configuration to NVRAM.

Appendix B: Acronyms

Acronym	Definition
AAA	Authentication, Authorization, and Accounting
AMI	Alternate Mark Inversion
ANSI	American National Standards Institute
APPN	Advanced Peer-to-Peer Networking
ARA	AppleTalk Remote Access Protocol
ARP	Address Resolution Protocol
ARPA	Advanced Research Projects Agency
ATM	Asynchronous Transfer Mode
AURP	Appletalk Update-based Routing Protocol
Bc	Committed Burst Size
Be	Excess Burst Size
BECN	Backward Explicit Congestion Notification
BERT	Bit Error Rate Tester
BFE	Blacker Front End
BGP	Border Gateway Protocol
BRI	Basic Rate Interface
BSC	Binary Synchronous Communications
BSTUN	Block Serial Tunnel
CCITT	Consultative Committee for International Telegraph & Telephone
CHAP	Challenge Handshake Authentication Protocol

CIP	Channel Interface Protocol
CIR	Committed Information Rate
CLI	Command Line Interface
CLNS	Connectionless Network Services
CMNS	Connection-Mode Network Service
CMNS	Connection Mode Network Services
CP	Control Point
CPU	Central Processor Unit
CSLIP	Compressed Serial Line Internet Protocol
CSNP	Complete Sequence Numer PDU
CSU	Channel Service Unit
CUD	Call User Packet
D-bit	Data Bit
DCA	Defense Communications Agency
DCE	Data-Circuit Terminating Equipment
DDN	Defense Data Network
DDN	Defense Data Network
DDR	Dial-on-Demand Routing
DDS	Digital Data Service
DE	Discard Eligibility
DHCP	Dynamic Host Configuration Protocol
DLCI	Data Link Connection Identifier
DLUR	Dependent LU Requester
DNIC	Data Network Identification Code
DS0	Digital Signaling 0
DSU	Digital Service Unit

DTE	Data Terminal Equipment
EGP	Exterior Gateway Protocol
ESF	Extended Superframe
FDDI	Fiber Distributed Data Interface
FIFO	First-In First-Out
FRMR	Frame Reject Frame
HDLC	High-level Link Control
HDLC	High-Level Data Link Control
HSRP	Hot Standby Router Protocol
HSSI	High-Speed Serial Interface
ICMP	Internal Control Message Protocol
IETF	Internet Engineering Task Force
IGRP	Internet Gateway Routing Protocol
IP	Internet Protocol
IPCP	IP Control Protocol
IPX	Internet Packet Exchange
IRDP	ICMP Router Discovery Protocol
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunications
L2F	Level 2 Forwarding
LAN	Local Area Network
LANE	LAN Emulation

LAPB	Link Access Procedure Balanced
LAPB	Link Access Procedure Balanced
LAPF	Link Access Procedure for Frame Relay
LAT	Local-Area Transport
LC	Logic Channel
LCI	Logic Channel Identifier
LEN	Low-Entry Networking
Level1	Station Router Level
Level2	Area Router Level
LLC2	Logical Link Control type 2
LMI	Local Management Interface
LQM	Link Quality Monitoring
LQR	Link Quality Reports
LSP	Link State PDU
LU	Logical Unit
MAC	Media Access Control
M-bit	More Data Bit
MBRI	Multiport BRI
MIP	MultiChannel Interface Processor
MLP	Multilink PPP
MMP	Multichasis Multilink PPP
MOP	Maintenance Operation Protocol
MTU	Maximum Transmission Limit
NAS	Network Access Server
NASI	Netware Asynchronous Services Interface
NCD	Network Control Device, Inc.

NCIA	Native Client Interface Architecture
NET	Network Entity Titles
NETID	Network Identifier
NLSP	Netware Link Services Protocol
NMP	Network Processor Module
NNI	Network-to-Network Interface
NSAP	Network Service Access Point
NVRAM	Nonvolatile Random Access Memory
OSI	Open System Interconnection
OSPF	Open Shortest Path First
PAD	Packet Assembler/Disassembler
PAP	Password Authentication Protocol
PDN	Public Data Network
PDU	Protocol Data Unit
PLP	Packet Level Protocol
POP	Point of Presence
POP	Point of Presence
PPP	Point-to-point Protocol
PRI	Primary Rate Interface
PSN	Packet-Switched Network
PSNP	Partial Sequence Number PDU
PU	Physical Unit
PVC	Permanent Virtual Circuits
QLLC	Qualified Logical Link Control
QOS	Quality of Service
RIP	Routing Information Protocol

RIP	Routing Information Protocol
ROM	Read Only Memory
RPOA	Recognized Private Operation Agency
RSRB	Remote Source-Route Bridging
RTMP	Routing Table Maintenance Protocol
SAP	Service Access Point
SDLC	Synchronous Data Link Control
SF	Superframe
SGBP	Stock Group Bidding
SLIP	Serial Line Internet Protocol
SMDS	Switched Multimegabit Data Service
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SR/TLB	Source-Route Translational Bridge
SRB	Source-Route Bridging
SRT	Source-Route Transport
SVC	Switched Virtual Circuit
TAC	Terminal Access Controller
TACACS	Terminal Access Controller Access Control System
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
THC	TCP/IP Header Compression
TOS	Type of Service

UDP	User Datagram Protocol
UP	Usage Parameter
URL	Universal Resource Locator
VCN	Virtual Circuit Nember
VPDN	Virtual Private Dial-up Network
VTY	Virtual Terminal
WAN	Wide Area Network
XNS	Xerox Network Systems
XOT	X.25 Over TCP