

Router Products Command Summary

Introduction

This booklet provides a summary of the commands a system administrator uses to configure a Cisco router for its routing and bridging tasks. The commands are listed in alphabetical order, by command type. The Table of Contents lists the section page numbers; the Index lists each command and the page it can be found on. See the *Router Products Configuration and Reference* for more complete descriptions and examples of the commands.

Conventions

The command descriptions use these conventions:

- Commands and keywords are in **boldface**.
- Variables for which you supply values are in *italics*.
- Elements in square brackets ([]) are optional.
- Alternative keywords are grouped in braces ({ }) and are separated by a vertical bar (|).
- The examples are printed in a `typewriter-type` font.
- Information you enter is in **boldface**.
- If the **no** form of the command is not explicitly explained in the description, it negates the command.

EXEC System Use

- Enter commands by typing their names at the EXEC prompt and pressing the Return key.
- There are two EXEC prompt levels. The user-level prompt is the server name followed by a >, like this example:

```
Router>
```

There is also a privileged-level prompt available to the system admini-

strator by entering a password. It is the server name followed by a #, like this example:

```
Router#
```

- Use the following editing commands when typing commands at the EXEC prompt:

Command	Function
Delete or Backspace	Erase characters
Ctrl-U	Delete Line

- As a shortcut, you can abbreviate commands to the fewest letters that make them unique. The letters “sho” can be entered for the **show** command, for example.
- Certain EXEC commands display multiple screens with this prompt at the bottom of the screen:

```
--More--
```

Type a space to continue the output; type any other key to return to the prompt.

System Help

- Type a ? to see a list of available commands at either EXEC prompt.
- You can type a ? after certain commands to get more information about them. See the command lists for more information.
- At any time during an active Telnet session, you can list the Telnet commands by typing this command at the system prompt:

Ctrl ^ ?

To enter the command above, press the Ctrl and Shift and 6 keys simultaneously, release them, and type a question mark (?).

The Configure Command

Use the privileged EXEC command **configure** to begin configuration of the router.

- Begin by entering the privileged level of the EXEC. To do this, type the **enable** command at the EXEC prompt:

```
Router>enable
```

The EXEC then prompts you for the privileged level password:

```
Password:
```

- Type the password, taking care to match uppercase and lowercase letters. For security purposes, the password does not appear. When you enter the correct password, the system displays the privileged-level prompt:

```
Router#
```

- To begin configuration mode, enter the **configure** command at the privileged-level prompt:

```
Router#configure
```

When you enter this command, the EXEC prompts you for the source of the configuration subcommands.

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

The default is to type in commands from the terminal console. Pressing the Return key begins this configuration method.

- The EXEC provides you with a simple editor for entering the configuration commands and explains the editing functions:

```
Enter configuration commands, one per line.  
Edit with DELETE, CTRL/W, and CTRL/U;end with CTRL/Z
```

The following table lists the edit key functions and their meanings.

Key	Meaning
Delete or Backspace	Erases one character.
Ctrl-W	Erases a word.
Ctrl-U	Erases a line.
Ctrl-R	Re-displays a line.
Return	Executes single-line commands.
Ctrl-Z	Ends configuration mode and returns to the EXEC.

Type Ctrl-Z to end configuration mode. Enter the **disable** EXEC command to return to the user-level EXEC prompt.

Configuration Command Types

Configuration commands are categorized by these functions:

- Global configuration commands—Define system-wide parameters.
- Interface subcommands—Define the characteristics of an interface (a serial or Ethernet interface, for example) and must be preceded by an **interface** command.
- Line subcommands—Define the characteristics of a serial line and must be preceded by a **line** command.
- Router subcommands—Specify a routing protocol (IGRP, OSPF, and so on) and must be preceded by a **router** command.

Observe the following guidelines when you enter configuration commands:

- As with EXEC commands, you can type configuration subcommands in uppercase letters, lowercase letters, or both. You may also shorten all commands and other keywords to the fewest number of characters that uniquely identify the word.
- To add a comment, begin the line with an exclamation point (!). Comments do not affect command processing.
- If the router encounters a problem, it displays an error message on the console terminal.

System EXEC Commands

EXEC Terminal Use Commands

{**connect** | **telnet**} [*connection*]

Connects to a remote host using the Telnet protocol.

connect or **telnet**—Use either of the two keywords.

connection—Host name or IP address.

Example:

```
connect router
```

disconnect [*connection*]

Closes the specified connection.

connection—Connection name or number displayed by the **show users** command; the default is the current connection.

Example:

```
disconnect 2
```

{**exit** | **quit**}

Either command terminates the EXEC command processor and closes any active Telnet sessions.

name-connection

Assigns a logical name to a connection. The EXEC prompts for the connection number and name to assign.

resume [*connection*]

Resumes a connection.

connection—Connection name or number.

Example:

```
resume 3
```

show sessions

Provides information about open Telnet connections.

show tcp [*line-number*]

Displays the status of all TCP connections, or, if the *line-number* argument is specified, displays the status of a single TCP connection.

line-number—Octal line number.

Example:

```
show tcp 0
```

show terminal [**all**]

Displays information about the terminal configuration parameter settings for the current terminal line and the active ports of the server.

all—Include this keyword to view information about inactive as well as active ports.

show users [**all**]**systat** [**all**]

Displays information about active lines.

all—Include this keyword to view information about inactive as well as active ports.

show version

Displays the configuration of the system hardware, the software version, the names and sources of configuration files, and the boot images.

where

Displays information about open connections associated with the current terminal line and provides the connection number.

EXEC Terminal Parameter Setting Commands

terminal ?

Lists commands you can enter to temporarily change hardware and software parameters of the current line.

terminal [no] escape-character *decimal-number*

Sets or removes the escape character for the current terminal line. Default: Ctrl ^ X.

decimal-number—The ASCII decimal representation of the desired escape character or an escape sequence.

Example:

```
terminal escape-character 17
```

terminal [no] length *screen-length*

Sets the terminal screen length. A screen length of zero or the **no** keyword disables pausing between screens of output. The screen length specified can be learned by hosts. Default: 24.

screen-length—Desired number of lines.

Example:

```
terminal length 0
```

terminal [no] monitor

Copies **debug** command output and system error messages to the current terminal as well as to the console terminal.

terminal [no] notify

Establishes or removes message notification.

terminal [no] padding *decimal-number count*

Sets or cancels character padding on the current terminal line.

decimal-number—ASCII decimal representation of the character.

count—Number of NULL bytes sent after that character.

Example:

```
terminal padding 25 20
```

terminal [no] terminal-type *terminal-name*

Records, removes, or changes the current terminal type.

terminal-name—Terminal name passed to applications that set terminal types.

Example:

```
terminal terminal-type VT100
```

terminal [no] width *columns*

Sets the number of characters (columns) on a single line of the current terminal screen. Default: 80.

column—Number of columns.

Example:

```
terminal width 132
```


EXEC System Management Commands

clear line *line-number*

Aborts connections and processes and resets a terminal line.

line-number—Terminal line number, displayed by **show user** command.

Example:

```
clear line 3
```

copy flash tftp

Copies a Flash TFTP image back to a TFTP server.

copy tftp flash

Copies a TFTP image into the current Flash configuration.

debug ?

Lists and briefly describes all the **debug** command options.

ping

Invokes a diagnostic tool for testing connectivity. Results are helpful for evaluating path-to-host reliability, delays over the path, and whether the host is functioning.

show ?

Lists **show** command options for the user-level prompt or privileged-level prompt, whichever is active.

show buffers [*interface*]

Displays statistics for the buffer pools on the network server.

interface—Interface name. When this argument is included, the router searches only those buffers that have been associated with the interface for longer than one minute.

Example:

```
show buffers Ethernet 0
```

show configuration

Displays the contents of nonvolatile memory.

show debugging

Displays the current settings of the **debug** command options.

show environment

Displays temperature and voltage information on the console.

show environment last

Displays the last measured value from each of six test points if the system shuts down due to detection of fatal environmental margins. The CSC-ENVM logs to internal nonvolatile memory. Only one set of measurements may be stored at any one time.

show flash [all]

Displays the amount of Flash memories available, the type of card connected to the Flash card, any files that may currently exist in Flash memory, and the amounts of Flash memory used and remaining.

all—Includes information about each Flash memory device.

show logging

Displays the state of syslog error and event logging, including host addresses and whether console logging is enabled. Also displays SNMP configuration parameters and protocol activity.

show memory

Displays memory free pool statistics, including summary information about the activities of the system memory allocator, and a block-by-block listing of memory use.

show processes

Displays information about all active processes.

show processes memory

Displays information about memory utilization.

show protocols

Displays the global and interface-specific status of any configured Level 3 protocol.

show stacks

Monitors the stack utilization of processes and interrupt routines. Displays the reason for the last system reboot and, if the system was reloaded because of a system failure, a saved system stack trace. This command is useful for analyzing system crashes.

test interfaces

Intended for use by Cisco technical personnel only; not for diagnosing problems with an operational router.

test memory

Performs a test of multibus memory, including nonvolatile memory.

test sbe

Tests CSC-R16 16Mbps Token Ring cards. Perform this test only at the direction of Cisco technical support, and do not issue the command while the system is in use.

trace [*destination*]

Allows the network administrator to discover the routes packets will actually take when traveling to their destination. Supports both IP and CLNS route tracing. To terminate the **trace** command operation, type the escape sequence.

undebug *option*

Disables diagnostic output enabled by **debug** command option.

Examples:

```
debug packet
undebug packet
```

write erase

Erases the configuration information in nonvolatile memory. This command does not affect the configuration in use.

write memory

Copies the current configuration information to nonvolatile memory.

write network

Sends a copy of the current configuration information to a server host. The system prompts for a destination host and a file name.

write terminal

Displays the current configuration information.

Global System and Interface Configuration Commands

[no] banner {motd | exec | incoming} c text c

Shows or removes the message that the EXEC command interpreter displays whenever a user starts any EXEC process or activates a line.

motd, exec, or incoming—Specifies when the banner message is displayed.

c—Delimiting character that you will use to signify the beginning and end of the banner message.

text—Message to be shown on the screen whenever an interface line is activated.

Example:

```
banner motd #
Building power will be off from 7:00 AM until 9:00 AM this
Tuesday.#
```

[no] boot buffersize bytes

Specifies the size of the buffer to be used for netbooting a host or a network configuration file. The **no** form of the command restores the default: size of your nonvolatile memory, or, if you do not have nonvolatile memory, 32 kilobytes.

bytes—Size of the buffer.

Example:

```
boot buffersize 64
```

[no] boot {host|network|system} filename [address]

Configures the system image boot files. The **no** form of the command with the appropriate keyword and filename removes the name. The command can be entered multiple times to build ordered lists.

host—Changes host configuration file name.

network—Changes network configuration file name.

system—Indicates that the file name and host address for the system image are in nonvolatile memory.

filename—New name for the host configuration file, or file name of operating system to load.

address—Network host or a subnet broadcast address.

Default: *255.255.255.255*.

Examples:

```
boot host config1
boot network config2
boot system opsoft
```

[no] boot system flash filename

Configures the system image boot files to boot the system from the Flash memories.

filename—File name of the image from the Flash memories.

Example:

```
boot system flash sysimage
```

[no] boot system rom

Configures the system to auto-boot from the ROM system image. This command is usually used as a backup to other **boot system** commands that specify system images existing on the network or in Flash memories.

[no] buffers {small | middle | big | large | huge}
{permanent / max-free / min-free / initial} number

Allows a network administrator to adjust initial buffer pool settings and set limits at which temporary buffers are created and destroyed. The **no** form of the command with appropriate keywords and arguments restores the default buffer values. Use this command only if instructed to by Cisco technical personnel.

small | middle | big | large | huge—Size of buffers in the pool. The default number is determined by the hardware configuration.

permanent / max-free / min-free / initial—Buffer management parameter to be changed.

number—Number of buffers to be allocated.

Example:

```
buffers small min-free 50
```

[no] buffers huge size number

Dynamically resizes all huge buffers to the value that you supply. The buffer size cannot be lowered below the default. The **no** version of the command with the argument restores the default buffer values. Use this command only when instructed to by Cisco technical personnel.

number—Number of buffers to be allocated.

Example:

```
buffers huge size 4
```

[no] dialer-list dialer-group list list-number
[no] dialer-list dialer-group protocol protocol-name {permit | deny}

Controls automatic dialing of DDR using standard IP or bridging access lists. Applies to dial-on-demand only.

dialer-group—Number of a dialer group identified in any **dialer group** interface subcommand.

list list-number—Access list number specified in any IP or bridging access list.

protocol *protocol-name*—One of the following supported protocols: **ip**, **decnet**, **chaos**, **xns**, **novell**, **appletalk**, **vines**, **apollo**, **pup**, **clns**, **bridge** (bridging), **stun** (serial tunneling), or **cmns** (OSI Connection-Oriented Network Service).

permit or **deny**—Specifies whether automatic dialing is permitted or denied.

Examples:

```
dialer-list 1 list 101
dialer-list 2 protocol apollo deny
```

enable password *password*

Assigns a password for the privileged command level.

password—Password prompted for in response to the EXEC command **enable**.

Example:

```
enable password yourpassword
```

[no] enable last-resort {succeed | password}

Allows you to specify what happens if the TACACS servers used by the **enable** command do not respond. The default action is to fail.

succeed or **password**—Use **succeed** to enable without further question. Use **password** to enable only if the user enters the privileged command level.

[no] enable use-tacacs

Enables or disables use of TACACS to check the user ID and password supplied to the EXEC **enable** command.

hostname *name*

Specifies the name for the network server. Default: Router.

name—Name of the network server.

Example:

```
hostname HAL
```

interface *type unit*

Specifies an interface and begins interface configuration.

type—Interface type: **hssi**, **serial**, **ethernet**, **tokenring**, **fdi**, or **ultranet**.

unit—Interface number or card number.

Example:

```
interface serial 0
```

line [*type-keyword*] *first-line* [*last-line*]

Identifies a specific line for configuration and starts line configuration command collection mode.

type-keyword—Type of line to be configured: **console**, **aux**, or **vt**.

first-line—If *type-keyword* is specified, provide the relative number (first line) in a contiguous group you want to configure. If *type-keyword* is not specified, provide the absolute number. Use the EXEC command **show users** to display line numbers.

last-line—If *type-keyword* is specified, provide the relative number (last line) in a contiguous group you want to configure. If *type-keyword* is not specified, provide the absolute number.

Example:

```
line vty 0 4
```

[no] logging *internet-address*

Identifies a syslog server host to receive logging messages.

internet-address—Internet address of the host.

Example:

```
logging 131.108.2.125
```

[no] logging buffered

Copies logging messages to an internal buffer instead of writing them to the console.

[no] logging console *level*

Limits the logging of messages displayed on the console terminal to messages at or above the specified level. Default: warnings.

The argument *level* is one of the following keywords:

emergencies—System unusable

alerts—Immediate action needed

critical—Critical conditions

errors—Error conditions

warnings—Warning conditions

notifications—Normal but significant conditions

informational—Informational messages only

debug—Debugging messages

[no] logging monitor *level*

Limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above *level*.

level—One of the keywords described for the **logging console** command, listed above.

Example:

```
logging monitor notifications
```

[no] logging on

Enables or disables message logging to all supported destinations except the console. Default: enabled.

[no] logging trap *level*

Limits the logging messages sent to syslog servers to messages with a level at or above *level*.

level—One of the keywords described for the **logging console** command.

Example:

```
logging trap errors
```

[no] priority-list *list default queue-keyword*

Assigns a priority queue for those datagrams that did not match any other rule in the priority list. If no default or the **no** form of the command is specified, the **normal** queue is assumed.

list—Integer from 1 to 10 that identifies the priority list selected by the user.

queue-keyword—Priority queue name: **high**, **medium**, **normal**, or **low**.

Example:

```
priority list 2 default medium
```

[no] priority-list *list interface interface-name queue-keyword*

Sets up priority queuing on the specified interface. The **no** form of the command removes the item from the list.

list—Integer from 1 to 10 that identifies the priority list selected by the user.

interface-name—Name of the interface.

queue-keyword—Priority queue name: **high**, **medium**, **normal**, or **low**.

Example:

```
priority list 1 interface Ethernet 2 medium
```

[no] priority-list *list protocol protocol-name queue-keyword [args]*

Sets up priority queuing by protocol type. The **no** form of the command removes the item from the list.

protocol-name—Name of the protocol: **ip, pup, chaos, xns, decnet, appletalk, clns, novell, apollo, vines, stun, bridge, rsrb.**

list—Integer from 1 to 10 that identifies the priority list selected by the user.

queue-keyword—Priority queue name: **high, medium, normal, or low.**

The optional keyword *args* is one of the following:

gt *byte-count*—Specifies a greater-than count. The priority level assigned goes into effect when a packet exceeds the value entered for the argument *byte-count*.

lt *byte-count*—Specifies a less-than count. The priority level assigned goes into effect when a packet size is less than the value entered for *byte-count*.

bridge list *list-number*—Assigns the priority level to bridged traffic according to the *list-number*, which is the Ethernet-type code access list number assigned using the **access-list** command.

list *list-number*—Assigns traffic priorities according to the *list-number*, which is the IP access list number assigned by the **access-group list** interface subcommand.

tcp *port*—Assigns the priority level defined to TCP packets originating from or destined to a specified port. Common TCP services and their port numbers appear in the *Router Products Configuration and Reference*.

udp *port*—Assigns the priority level defined to UDP packets originating from or destined to the specified port. Common UDP services and their port numbers appear in the *Router Products Configuration and Reference*.

Examples:

```
priority-list 1 protocol decnet high
priority-list 4 protocol decnet medium lt 200
priority-list 4 protocol ip medium tcp 23
```

[no] priority-list *list queue-limit high-limit medium-limit normal-limit low-limit*

Specifies the maximum number of packets that can wait in a single priority queue. If a priority queue overflows, the router discards excess datagrams and may send quench messages. The **no** form of the command resets all four queue sizes to their defaults: *high-limit* = 20; *medium-limit* = 40; *normal-limit* = 60; *low-limit* = 80.

list—Integer from 1 to 10 that identifies the priority list selected by the user.

Example:

```
priority-list 1 queue-limit 20 20 20 10
```

[no] scheduler-interval *milliseconds*

Sets the maximum amount of time that can elapse without the router running the lowest priority system processes. The minimum interval that may be specified is 500 milliseconds; there is no maximum value. The **no** form of the command restores the default: no maximum.

milliseconds—Number of milliseconds. Must be 500 or greater.

Example:

```
scheduler-interval 750
```

[no] service *keyword*

Tailors use of network-based services by the network server.

The argument *keyword* is one of the following:

config—Specifies TFTP autoloading of configuration files; disabled by default on systems with nonvolatile memory.

decimal-tty—Specifies that line numbers be displayed and interpreted as decimal numbers rather than octal numbers; disabled by default.

finger—Allows Finger protocol requests (defined in RFC 742) to be made of the network server; enabled by default.

tcp-keepalives-*{in | out}*—Generates keepalive packets on idle network connections. The **in** keyword generates them on incoming connections; the **out** keyword generates them on outgoing connections.

Example:

```
no service config
```

[no] smt-queue-threshold *number*

Sets or removes the maximum number of unprocessed FDDI Station Management (SMT) frames that the router will hold for processing. The **no** form of the command restores the default, which is the number of installed FDDI interfaces.

number—Positive integer.

Example:

```
smt-queue-threshold 2
```

no snmp-server

Disables the SNMP operations.

[no] snmp-server access-list *list*

Sets up an access list that determines which hosts can send requests to the network server. Applies only to the global read-only SNMP agent configured by the **snmp-server community** command.

list—IP list, expressed as an integer from 1 to 99.

Example:

```
snmp-server access-list 20
```

snmp-server community [*string* [**RO** | **RW**] [*list*]]

no snmp-server [**community** [*string*]]

Enables or disables SNMP server operation on the network server.

string—Community string that acts like a password and permits access to the SNMP protocol.

RO or **RW**—The **RO** keyword specifies read-only access (default); the **RW** keyword specifies read-write access.

list—Integer from 1 through 99 that specifies an access list of Internet addresses that can use the community string.

Example:

```
snmp-server community yourstring RO 4
```

[no] snmp-server host *address community-string [snmp | tty]*

Specifies which host or hosts should receive TRAP messages.

address—Name or Internet address of the host.

community-string—Community string, similar to a password, that was set with the **snmp-server community** command.

The keywords **snmp** or **tty** specify the TRAP type, as follows:

snmp—Sends all SNMP-type TRAP messages and starts the Cisco-specific RELOAD TRAP message.

tty—Includes TCP connection TRAP messages.

Example:

```
snmp-server host 131.108.2.160 yourstring
```

[no] snmp-server packetsize *bytes*

Sets or removes control over the largest SNMP packet size permitted when the SNMP server is receiving a request or generating a reply.

bytes—Byte count from 484 to 8192. Default: 484.

Example:

```
snmp-server packetsize 8192
```

[no] snmp-server queue-length *length*

Defines the length of the message queue for each TRAP host. Default: 10.

length—Number of TRAP events that can be held before the queue must be emptied.

Example:

```
snmp-server queue-length 4
```

[no] snmp-server system-shutdown

Allows or restricts use of the SNMP message reload feature. Prevents an SNMP system-shutdown request from resetting the Cisco agent.

[no] snmp-server trap-authentication

Allows the network server to send a TRAP message when it receives a packet with an incorrect community string. The **no** form of the command restricts the sending of TRAP messages.

[no] snmp-server trap-timeout *seconds*

Defines how often the router attempts to resend TRAP messages in the retransmission queue. The **no** form of the command restores the default: 30 seconds.

seconds—Number of seconds in the interval.

Example:

```
snmp-server trap-timeout 20
```

[no] tacacs-server attempts *count*

Controls the number of login attempts that may be made on a line set up for TACACS verification. The **no** form of the command allows no attempts. Default: 3.

count—Number of attempts.

Example:

```
tacacs-server attempts 6
```

[no] tacacs-server authenticate {connect|enable}

Specifies that a response is required from the network or communications server before a user can perform a specific action.

The **no** form of the command removes the response requirement.

Select the action that requires a response:

connect—Make TCP connections.

enable—Enter the **enable** command.

[no] tacacs-server extended

Enables or disables an extended TACACS mode.

[no] tacacs-server host *name*

Specifies a TACACS host.

name—Name or Internet address of the host.

Example:

```
tacacs-server host host1
```

[no] tacacs-server last-resort {password|succeed}

Causes the network server to request the privileged password as verification, or forces successful login without further input from the user. The **no** form of the command removes the specification.

Select one keyword to configure the desired action:

password—Allows the user to access the privileged-level command mode by entering the password set by the **enable** command.

succeed—Allows the user to access the privileged-level command mode without further question.

tacacs-server notify {connect | enable | logout}

Causes a message to be transmitted to the TACACS server; the message is retransmitted in the background for up to five minutes. The **no** form of the command removes the specification.

Select one keyword to specify when the TACACS server is notified:

connect—Make TCP connections.

enable—Enter the **enable** command

logout—Log out

tacacs-server optional-passwords

Specifies that the first TACACS request to a TACACS server is made *without* password verification.

[no] tacacs-server retransmit *retries*

Specifies the number of times the server will search the list of TACACS server hosts before abandoning the attempt. The **no** form of the command restores the default of 2.

retries—Retransmit count.

Example:

```
tacacs-server retransmit 4
```

[no] tacacs-server timeout *seconds*

Sets the interval the server waits for a server host to reply. The **no** form of the command restores the default of 5 seconds.

seconds—Number of seconds.

Example:

```
tacacs-server timeout 10
```

[no] tftp-server system *filename ip-access-list*

Specifies or removes TFTP server operation for a communications server.

filename—Name of the network server ROM file.

access-list—IP access list number.

Example:

```
tftp-server system configfile 22
```

username *name* [**nopassword** | **password** *encryptiontype password*]

username *name* [**accesslist** *number*]

username *name* [**autocommand** *command*]

username *name* [**noescape**] [**nohangup**]

Implements a user name-based authentication system for networks that cannot support a TACACS service. Also defines user names that get special treatment.

nopassword—No password is required for this user to log in.

password—Specifies a possibly encrypted password for this user name.

encryptiontype—Single digit number. Currently defined encryption types are zero, which means no encryption, and seven, which specifies a Cisco-defined encryption algorithm.

accesslist—Specifies an outgoing access list that overrides the access list specified in the **access class** line configuration subcommand.

autocommand—Causes the command specified by the *command* argument to be issued automatically after the user logs in.

nohangup—Prevents the network server from disconnecting the user after an automatic command.

noescape—Prevents user from having an escape character with which to escape from the host system.

Examples:

```
username who nopassword nohangup autocommand show users
username superuser password yourpassword
```

Line Configuration Subcommands

[no] access-class *list* {**in** | **out**}

Restricts or permits connections on a line or group of lines to certain Internet addresses.

list—Integer from 1 to 99 that identifies a specific access list of Internet addresses.

in or **out**—The keyword **in** applies to incoming Telnet connections; the keyword **out** applies to outgoing Telnet connections.

Example:

```
access-class 99 in
```

[no] escape-character *decimal-number*

Sets or removes the escape character on the specified line. Default escape character: Ctrl ^ X.

decimal-number—ASCII decimal representation of the character or a control sequence.

Example:

```
escape-character 13
```

[no] exec-banner

Enables or disables a banner. Default: enabled.

[no] exec-timeout *minutes* [*seconds*]

Sets the interval the EXEC waits for user input before resuming the current connection, or if no connections exist, before returning the terminal to the idle state and disconnecting the incoming session. The **no** form of the command is the same as specifying a timeout of 0.

minutes—Number of minutes. The default interval is ten minutes

seconds—Number of seconds; an interval of zero specifies no time outs.

Example:

```
exec-timeout 15 30
```

[no] length *screen-length*

Sets the terminal screen length. A screen length of zero disables pausing between screens of output. Default: 24.

screen-length—Number of lines on the screen.

Example:

```
length 32
```

[no] location *text*

Enters or removes textual description concerning the terminal location and/or status.

text—Desired description.

Example:

```
location In the hall
```

[no] login

Enables or disables checking for the password specified by the **password** command.

[no] login tacacs

Invokes use of the TACACS user ID and password-checking mechanism instead of regular password checking. The **no** form of the command disables this mechanism.

[no] notify

Enables or disables line notification when a user running multiple, concurrent Telnet connections has output pending on a connection other than the current line.

[no] padding *decimal-number count*

Sets or cancels character padding for a specified output character.

decimal-number—ASCII decimal representation of the character.

count—Number of NULL bytes sent after the character.

Example:

```
padding 13 25
```

[no] password *text*

Specifies a password.

text—Password containing up to 80 alphanumeric characters, including spaces.

Example:

```
password yourpassword
```

vacant-message [*c message c*]

no vacant-message

Controls whether or not a banner is displayed on the screen of an idle terminal. The command without any arguments causes the default message to be displayed. The **no vacant-message** command suppresses a banner message.

c—Delimiting character that you will use to signify the beginning and end of the banner message.

message—Message to be shown on the screen whenever an interface line is activated.

Example:

```
vacant-message #  
                Welcome to Cisco Systems, Inc.  
                #
```

Interface and Media Configuration

Interface Configuration Subcommands

[no] backup delay {*enable-delay*|**never**} {*disable-delay*|**never**}

Defines how much time should elapse before a line is set up or taken down.

enable-delay—Number of seconds after the primary line goes down that the secondary line is activated. The **never** keyword prevents the line from ever coming up due to primary going down,

disable-delay—Number of seconds after the primary line goes up that the secondary line is deactivated. The **never** keyword prevents the line from ever going down due to primary coming up.

Examples:

```
backup delay 10 never
backup delay 30 60
```

[no] backup interface *interface-name*

Configures the serial interface as a secondary line (dial backup line). The **no** form of the command removes this configuration.

interface-name—Serial port to be set as the secondary interface line.

Example:

```
backup interface serial 1
```

[no] backup load {*enable-threshold* |**never**} {*disable-load* |**never**}

Sets the traffic load thresholds. The **no** form of the command restores the default: no backup loads.

enable-threshold and *disable-load*—Numbers representing the transmitted load as a percentage of the primary line's available bandwidth.

never—Prevents the secondary line from being enabled or disabled due to load.

Example:

```
backup load 60 5
```

[no] description *name-string*

Adds a descriptive name to an interface.

name-string—Comment displayed in the configuration file.

Example:

```
description 3174 Controller for test lab
```

[no] dialer enable-timeout *number-of-seconds*

Sets the length of time an interface stays down before it is available for dialing. The **no** form of the command restores the default of 15 seconds.

number-of-seconds—Number of seconds an interface stays down before it is available to dial again.

Example:

```
dialer enable-timeout 10
```

[no] dialer fast-idle *number-of-seconds*

Specifies the idle time before the line is disconnected for interfaces for which there is an unusually high level of contention. The **no** form of the command restores the default of 20 seconds.

number-of-seconds—Number of seconds.

Example:

```
dialer fast-idle 30
```

[no] dialer-group *group-number*

Assigns an interface to a set of access list expressions. These access list expressions define which packets cause a connection to be established and which keep an interface from being idle. Applies to dial-on-demand only.

The **no** form of the command removes an interface from the specified dialer-group.

group-number—Number of the dialer group to which the specific interface belongs. This group number corresponds to a dialer group defined using the **dialer-list** command.

Example:

```
dialer-group 3
```

[no] dialer idle-timeout *number-of-seconds*

Specifies the idle time before the line is disconnected. Applies to dial-on-demand only. The **no** form of the command restores the default of 120 seconds.

number-of-seconds—Number of seconds of idle time that must occur on an interface before the line is disconnected.

Example:

```
dialer idle-timeout 180
```

[no] dialer in-band

Sets or removes v.25 bis dialing for the specified serial interface.

[no] dialer map *protocol next-hop-address dial-string*

Defines multiple dial-on-demand numbers for a particular interface. Applies to dial-on-demand only. The **no** form of the command deletes a particular dialer map entry.

protocol—Supported protocol; currently must be **ip**.

next-hop-address—IP address used to compare against addresses to which packets are destined.

dialer-string—Telephone number sent to the DCE dialing device when packets with the specified *next-hop-address* are seen.

Example:

```
dialer map ip 131.108.2.5 14155553434
```

[no] dialer string *dial-string*

Specifies or deletes the telephone number to be passed to the DCE device, typically a v.25 bis modem.

dial-string—Character string to be passed to the v.25 bis DCE device.

Example:

```
dialer string 14155553434
```

[no] dialer wait-for-carrier-time *number-of-seconds*

Specifies how long the router will wait for carrier. The **no** form of the command restores the default: 30 seconds.

number-of-seconds—Number of seconds.

Example:

```
dialer wait-for-carrier-time 5
```

encapsulation *encapsulation-type*

Assigns an encapsulation method.

The argument *encapsulation-type* identifies a supported encapsulation type:

arpa—Ethernet version 2.0 encapsulation

bfex25—Blacker Front End Encryption X.25 operation

ddnx25-dce—DDN X.25 DCE operation

ddnx25—DDN X.25 DTE operation

frame-relay—Frame relay

hdh—HDH protocol

hdlc—HDLC protocol

iso1—IEEE 802.3 encapsulation

lapb-dce—X.25 LAPB DCE operation

lapb—X.25 LAPB DTE operation

multi-lapb-dce—X.25 LAPB multiprotocol DCE operation

multi-lapb—X.25 LAPB multiprotocol DTE operation

ppp—Point-to-Point Protocol (PPP)

snap—IEEE 802.2 Ethernet media

smds—SMDS service

x25-dce—X.25 DCE operation

x25—X.25 DTE operation

[no] early-token-release

Enables or disables the ability of Token Ring interfaces to release the token to the ring immediately after transmitting. Default: disabled.

fddi cmt-signal-bits *signal-bits* {**phy-a** | **phy-b**}

Controls the information transmitted during the FDDI CMT signaling phase.

signal-bits—Hexadecimal number preceded by “0x.”

phy-a or **phy-b**—Physical Sublayer: Physical A or Physical B station.

Example:

```
fddi cmt-signal-bits 0x208 phy-a
```

fddi tl-min-time *microseconds*

Controls the FDDI TL_MIN time, which is the minimum time to transmit a Physical Sublayer, or PHY line state, before advancing to the next Physical Connection Management, or PCM state.

microseconds—Usually 30; some implementations of the FDDI standard need more than 30 microseconds to sense a signal.

Example:

```
fddi tl-min-time 30
```

fddi token-rotation-time *microseconds*

Controls ring scheduling during normal operation, and detects and recovers from serious ring error situations. Default: 5000 microseconds.

microseconds—Number of microseconds in the token rotation time (TRT).

Example:

```
fddi token-rotation-time 2500
```

fddi valid-transmission-time *microseconds*

Sets the transmission valid timer (TVX) interval. Default: 2,500 microseconds.

microseconds—Number of microseconds.

Example:

```
fddi valid-transmission-time 1250
```

[no] keepalive [*seconds*]

Adjusts the keepalive timer for a specific interface. Default: 10 seconds.

seconds—Number of seconds after which the keepalive timer is activated.

Example:

```
keepalive 3
```

ring-speed *speed*

Sets operational ring speed for Token Ring interface. Default: 16.

speed—Operation speed: 4 or 16 Mbps.

Example:

```
ring-speed 4
```

[no] shutdown *interface*

Disables or enables an interface.

interface—Interface name.

Example:

```
shutdown ethernet0
```

ultranet address *ultranet-mac-address*

Assigns the MAC layer address of the interface.

ultranet-mac-address—MAC address of the UltraNet service line.

Example:

```
ultranet address 8/32
```

Interface Configuration EXEC Commands

clear counters [*type unit*]

Clears all the current interface counters from the interface, or clears those counters described by the optional arguments.

type—Interface type: **serial**, **ethernet**, **tokenring**, **fddi**, **hssi**, or **ultranet**.

unit—Interface unit or card number.

Example:

```
clear interface serial 1
```

clear interface *type unit*

Resets the hardware logic on an interface.

type—Interface type: **serial**, **ethernet**, **tokenring**, **fddi**, **hssi**, or **ultranet**.

unit—Interface connection or card number.

Example:

```
clear interface fddi 0
```

cmt connect [*interface* [**phy-a** | **phy-b**]]

Starts the FDDI CMT process.

interface—FDDI interface.

phy-a or **phy-b**—Physical Sublayer A or B.

Example:

```
cmt connect fddi 0 phy-a
```

cmt disconnect [*interface* [**phy-a** | **phy-b**]]

Stops the FDDI CMT process.

interface—FDDI interface.

phy-a or **phy-b**—Physical Sublayer A or B.

Example:

```
cmt disconnect fddi 0 phy-a
```

show controller {**serial** | **token** | **mci** | **cbus** | **fddi**}

Displays current internal status information for different interface cards.

show interfaces [*type unit*]

Displays statistics for the network interfaces on the network server.

type—Interface type: **ethernet**, **hssi**, **serial**, **tokenring**, **ultranet**, or **fddi**.

unit—Interface unit or card number.

Example:

```
show interfaces fddi 0
```

Interface Adjustment Subcommands

[no] bandwidth *kilobits*

Sets a bandwidth value for an interface. This is a routing parameter only; it does not affect the physical interface. The **no** form of the command restores the default, which is set during start up.

kilobits—Bandwidth in kilobits per second.

Example:

```
bandwidth 45045
```

[no] clockrate *speed*

Sets the clock rate on the serial interface of the SCI and MCI cards. The **no** form of the command removes the command from the configuration.

speed—Clock rate, in bits per second.

Example:

```
clockrate 64000
```

[no] delay *tens-of-microseconds*

Sets the delay that higher level protocols can use to make operating decisions. The **no** form of the command restores the default: no delay.

tens-of-microseconds—Delay for an interface or network segment, expressed in tens of microseconds.

Example:

```
delay 20
```

error-threshold *milliseconds*

Sets the frequency at which the error recount will be set. This protects against packet overload and resultant recount errors on the MCI interface cards. Default: 1000 milliseconds.

milliseconds—Frequency, in milliseconds.

Example:

```
error-threshold 10000
```

[no] hold-queue *length* {**in** | **out**}

Specifies the hold-queue limit of an interface. The **no** form of the command restores the default values for the interface.

length—Maximum number of packets in the queue.

in or **out**—Use **in** to specify the input queue; use **out** to specify the output queue.

Example:

```
hold-queue 40 in
```

[no] hssi external-loopback-request

Enables or disables an external loopback request on HSSI from the CSU/DSU.

[no] loopback

Achieves the following affect, depending on the interface:

- HSSI—Configures a two-way internal and external loopback loop on the HSA applique of the specified interface.
- UltraNet—Configures a two-way, internal, and external loopback on the ULA applique of the specified interface.
- MCI and SCI serial cards—Loops the packets through the CSU/DSU to configure a “CSU loop,” when the device supports this feature. This is similar to the **loopback line** command on the HSSI.

- MCI and MEC Ethernet cards—Loops the packets at the interface within the router.
- CSC-R16 card—Loops the packets at the interface within the router.

The **no** form of the command disables the loopback test.

[no] loopback applique

For use with HSSI connections only. Configures an internal loop on the HSSI. This command is useful for sending pings to yourself to check functionality of the applique. Equivalent to the **loopback** command for HSSI.

[no] loopback dte

For use with HSSI connections only. Checks connectivity between the router and the CSU/DSU by looping packets from the router to the CSU/DSU and back.

[no] loopback line

For use with HSSI connections only. Tests functionality of the CSU/DSU by looping the packets through the CSU/DSU to configure a CSU loop.

[no] loopback remote

For use with HSSI connections only. Loops the packets through the CSU/DSU and over the DS3 link to the remote CSU/DSU.

[no] mtu bytes

Adjusts the default maximum transmission unit (MTU) size. The **no** form of the command restores the default for the interface.

bytes—Desired size, in bytes. Must be at least 64 bytes for serial interfaces.

Example:

```
mtu 576
```

priority-group *list*

Assigns a priority group to an interface.

list—Priority list number assigned to the interface.

Example:

```
priority-group 1
```

[no] pulse-time *seconds*

Enables or disables pulsing DTR signals on the Cisco MCI and SCI serial interfaces for a minimum interval.

seconds—Number of seconds in the minimum interval.

Example:

```
pulse-time 3
```

[no] transmitter-delay *microseconds*

For MCI and SCI serial cards only. Specifies a minimum dead-time after the router transmits a datagram. The **no** form of the command restores the default: zero microseconds.

microseconds—Approximate number of microseconds.

Example:

```
transmitter-delay 300
```

[no] transmitter-delay *hdlc-flags*

For the Cisco IGS router and the High Speed Serial Interface (HSSI) connector only. Specifies the minimum number of HDLC flags that the router should send between packets. The **no** form of the command restores the default: zero seconds.

hdlc-flags—Number of HDLC flags. Valid range on IGS is 2 to 62; valid range on HSSI is 2 to 128000.

Example:

```
transmitter-delay 40
```

Frame Relay Interface Subcommands

encapsulation frame-relay

Specifies frame-relay encapsulation on a specific interface.

[no] frame-relay keepalive *seconds*

Enables and disables the LMI mechanism for serial lines using the frame relay encapsulation. Default: 10 seconds.

seconds—Keepalive interval, in seconds.

Example:

```
frame-relay keepalive 15
```

frame-relay local-dlci *number*

Used for testing. Sets the source data link connection identifier (DLCI) for use when the local management interfaces (LMI) is not supported. If LMI is supported and the multicast information element is present, the network server sets its local DLCI based on information provided via the LMI.

number—Local, or source, DLCI number.

Example:

```
frame-relay local-dlci 100
```

[no] frame-relay lmi-type ANSI

Specifies the exchange of local management interface messages as defined by ANSI standard T1.617. The **no** form of the command restores the LMI type to the default as defined by the Cisco/StrataCom/Northern Telecom/DEC specification.

[no] frame-relay map *protocol protocol-address DLCI* [**broadcast**]
frame-relay map bridge *DLCI broadcast*
frame-relay map clns *DLCI broadcast*
no frame-relay map

Defines the mapping between an address and the DLCI used to connect to the address. The frame relay map tells the network server how to get from a specific protocol and address pair to the correct DLCI. The **no** form of the command deletes the mapping entry.

bridge—Specifies bridging.

clns—Specifies ISO CLNS protocol.

protocol—One of these keywords: **ip**, **decnet**, **appletalk**, **xns**, **novell**, **vines**, or **clns**.

protocol-address—Protocol address.

DLCI—DLCI number.

broadcast—Specifies that broadcasts should be forwarded to this address when the multicast is not enabled.

Examples:

```
frame-relay map IP 131.108.123.1 100
frame-relay map bridge 144 broadcast
```

frame-relay multicast-dlci *number*

Defines a DLCI to be used for multicasts. Use this command only when the multicast facility is *not* supported. Network transmissions (packets) sent to a multicast DLCI are delivered to all network servers defined as members of the multicast group.

number—Multicast group number.

Example:

```
frame-relay multicast-dlci 1022
```

[no] frame-relay short-status

Instructs the network server to request the short status message from the switch (see Version 2.3 of the joint *Frame Relay Interface* specification).
Default: disabled (full status message).

SMDS Global Command

arp *ip-address address smds*

Allows inclusion of static ARP entries.

ip-address—IP address.

address—SMDS address.

Example:

```
arp 128.27.101.8 c141.5797.1311 smds
```

SMDS Interface Subcommands

encapsulation smds

Enables or disables SMDS on a particular interface. This command must precede all SMDS commands.

[no] smds address *smds-address*

Sets or removes the SMDS individual address for a particular interface.

smds-address—Individual address provided by the SMDS service provider.

Example:

```
smds address C141.5797.1313
```

[no] smds d15-mode

Enables or disables AT&T's d15 packet structure. Default: enabled.

[no] smds enable-arp

Enables or disables ARP processing on a particular interface. Set the multicast address for ARP before issuing this command. Default: disabled.

[no] smds multicast *protocol-type smds-group-address*

Maps an SMDS group address to a broadcast or multicast address used by higher-level protocols. The **no** form of the command, with arguments, removes a multicast address.

protocol-type—Name of the protocol: **ip, arp, decnet, decnet_router, decnet_node, appletalk, aarp, xns, novell, clns, clns_is, or clns_es.**

smds-group-address—SMDS address.

Example:

```
smds multicast IP E180.0999.9999
```

[no] smds static-map *protocol-type protocol-address smds-address*

Sets up a static mapping between an SMDS address and a higher-level protocol address. Do not use this command for broadcast addresses; instead, use the **smds multicast** command.

protocol-type—Name of the protocol: **ip, decnet, appletalk, xns, novell, or clns.**

protocol-address—Address of specified protocol.

smds-address—SMDS address.

Example:

```
smds static-map XNS 111.00C0.2711.0123 C141.5688.1212
```

X.25 Global Configuration Commands

[no] **x25 route** [#position]x121-patt [**cu**d patt] **interface** int-name
[no] **x25 route** [#position]x121-patt [**cu**d patt] **ip** ip-address
[no] **x25 route** [#position]x121-patt [**cu**d patt] **alias** int-name
[no] **x25 route** [#position]x121-patt [**sub**stitute-source pattern]
[**sub**stitute-dest pattern] [**cu**d patt] **interface** destination

Inserts or removes an entry in the X.25 routing table.

#position—Positional parameter that indicates the existing entry after which the current entry will be inserted or deleted.

x121-patt—X.121 address of the called destination.

cud patt—Call User Data and pattern, or regular expression that must match the called address.

interface int-name—Interface name.

ip ip-address—IP address.

alias—Treats X.121 addresses as local.

substitute-source—Allows substitution of the calling address.

substitute-dest—Allows substitution of the destination address.

pattern—Replaces the called or calling X.121 address in routed X.25 packets.

interface destination—Destination address of the specified interface.

Example:

```
x25 route ^1111(.*) substitute-dest 2222\1 interface  
serial 3
```

[no] x25 routing

Enables or disables X.25 switching. X.25 calls are not forwarded until this command is issued.

LAPB and X.25 Interface Subcommands

[no] cmns enable

Enables or disables CMNS on a nonserial interface.

hdh {packet | message}

Enables the packet or message mode of the HDH protocol.

packet—Specifies the packet mode.

message—Specifies the message mode.

lapb k *window-size*

Defines the maximum permissible number of outstanding LAPB frames.

window-size—Packet count, from 1 to 7.

Example:

```
lapb k 4
```

lapb n1 *bits*

Defines the maximum number of bits a LAPB frame can hold. Default: 12000 bits (5000 bytes).

bits—Number of bits. Must be a multiple of 8 from 1 to 16384.

Example:

```
lapb n1 6000
```

lapb n2 *retries*

Defines the maximum number of times a LAPB acknowledgment frame can be retransmitted. Default: 20.

retries—Retransmission count, from 1 to 255.

Example:

```
lapb n2 100
```

lapb t1 *milliseconds*

Defines the length of time a LAPB transmitted frame can remain unacknowledged before the router polls for an acknowledgment. Default: 3000.

milliseconds—Number of milliseconds, from 1 to 64000.

Example:

```
lapb t1 32000
```

[no] x25 accept-reverse

Instructs the router to accept all reverse charge calls. This behavior can also be configured on a peer-to-peer basis using the **x25 map** subcommand. The **no** form of the command restore the default: rejection of reverse charge calls.

x25 address *X.121-address*

Sets the X.121 address of a particular network interface. The address is assigned by the X.25 network supplier.

X.121-address—Variable-length X.121 address.

Example:

```
x25 address 0000.0c00.ff89
```

[no] x25 default [ip | pad]

Specifies or removes a protocol by which the router interprets calls with unknown Call User Data. The protocol is either **ip** or **pad**.

[no] x25 facility *keyword argument*

Overrides interface settings on a per-call basis. This enables X.25 facilities that are sent between DTE and DCE devices to negotiate certain link parameters. The **no** form of the command, with keyword and argument, removes the facility.

Choose one of the following keyword-argument pairs:

cug number—Specifies a Closed User Group *number* from 1 through 99 to provide an extra measure of network security.

packetsize in-size out-size—Sets the size in bytes of input packets (*in-size*) and output packets (*out-size*). Both values should be the same.

reverse—Reverses charges on all Call Request packets from the interface.

window-size in-size out-size—Sets the packet count for input windows (*in-size*) and output windows (*out-size*). Both values should be the same.

throughput in out—Sets the requested throughput values for input and output throughput across the network.

rpoa name—Specifies the list of transit Recognized Private Operating Agencies (RPOAs) to use in outgoing Call Request packets.

transit-delay number—Specifies the transit delay value in milliseconds (0 to 65334) for the mapping in of outgoing calls.

Example:

```
x25 facility cug 1
x25 facility transit-delay 10
```

x25 hic channel

Sets the highest incoming channel (HIC). Default: 1024.

channel—Channel number, from 1 to 4095.

Example:

```
x25 hic 2048
```

x25 hoc channel

Sets the highest outgoing channel (HOC). Default: 1024.

channel—Channel number, from 1 to 4095.

Example:

```
x25 hoc 2048
```

[no] x25 hold-queue *queue-size*

Defines the number of packets the router can hold until a virtual circuit is established. The **no** form of the command restores the default of zero.

queue-size—Number of packets.

Example:

```
x25 hold-queue 3
```

[no] x25 hold-vc-timer *minutes*

Prevents overruns on X.25 switches for traffic through the virtual circuits (VCs) for a specified period. When this command is activated, incoming calls are still accepted. Default: zero.

minutes—Number of minutes to prevent calls to a previously failed destination.

Example:

```
x25 hold-vc-timer 1
```

x25 htc *channel*

Sets the highest two-way channel (HTC). Default: 1024.

channel—Channel number, from 1 to 4095.

Example:

```
x25 htc 512
```

[no] x25 idle *minutes*

Sets the period of inactivity once an SVC has been cleared. Affects calls both originated and terminated by the router. The **no** form of the command restores the default of zero minutes.

minutes—Number of minutes in the period. Both calls originated and terminated by the router are cleared.

Example:

```
x25 idle 1
```

[no] x25 ip-precedence

Enables or disables the router's ability to open a new virtual circuit based on the IP Type of Service (TOS) field. By default, Cisco routers open one virtual circuit for each type of service.

x25 ips bytes

Sets the router input packet size to match those of the network. Default: 128 bytes.

byte—Byte count from 128 to 1024. Larger packet sizes are better. Use the same byte value for **x25 ips** and **x25 ops** unless your network supports asymmetry between input and output packets.

Example:

```
x25 ips 1024
```

x25 lic channel

Sets the lowest incoming channel (LIC). Default: 1.

channel—Channel number from 1 to 4095.

Example:

```
x25 lic 2
```

[no] x25 linkrestart

Forces a packet-level restart when the link level is restarted. Restarts X.25 Level 2 (LAPB) when errors occur. The **no** form of the command turns off this behavior. Default: enabled.

x25 loc channel

Sets the lowest outgoing channel (LOC). Default: 1.

channel—Channel number, from 1 to 4095.

Example:

```
x25 loc 2
```

x25 ltc channel

Sets the lowest two-way channel (LTC). Default: 1.

channel—Channel number, from 1 to 4095.

Example:

```
x25 ltc 2
```

[no] **x25 map** *protocol-keyword protocol-address X.121-address [option1...option6]*

Specifies a protocol-to-X.121 address mapping, such as Internet-to-X.121 or DECnet-to-X.121.

protocol-keyword—Protocol type: **ip, decnet, chaos, xns, novell, appletalk, vines, apollo, pup, clns, bridge, or cmns.**

protocol-address—Protocol address to be mapped.

X.121-address—X.121 address to which the protocol address will be mapped.

Choose up to six of the following mapping features, in the order listed, for the *option* argument:

reverse—Specifies reverse charging for outgoing calls.

accept-reverse—Instructs the router to accept incoming reverse-charged calls. If this option is not present, the router clears reverse charge calls.

broadcast—Instructs the router to direct any broadcasts sent through this interface to the specified X.121 address. This option is needed when dynamic routing protocols are being used to access the X.25 network.

cug number—Specifies a Closed User Group number (from 1 to 99) for the mapping in the outgoing call.

nvc count—Sets the number of virtual circuits (VCs) for this map/host. The default *count* is the **x25 nvc** setting of the interface. A maximum number of eight VCs can be configured for a single map/host.

packetsize *in-size out-size*—Specifies input packet size *in-size* and output packet size *out-size* for the mapping in the outgoing call.

window size *in-size out-size*—Specifies input window size *in-size* and output window size *out-size* for the mapping in the outgoing call.

throughput *in out*—Requests the amount of bandwidth through the X.25 network.

modulo *size*—Specifies the maximum window size for this map. The argument *size* permits windows of 8 or 128 on the same interface.

transit-delay *number*—Specifies the transit delay value in milliseconds (0 to 65334) for the mapping in of outgoing calls. Used only for networks that support delay transit.

nuid *username password*—Specifies that a network ID facility be sent in the outgoing call with the specified user name and password.

Example:

```
x25 map ip 131.108.9.2 31370054065 reverse broadcast nvc 4
```

x25 map bridge *X.121-address* **broadcast** [*options-keywords*]

Specifies Internet-to-X.121 mapping.

X.121-address—X.121 address.

options-keywords—Services that can be added to this map. See the **x25 map** command for options.

Example:

```
x25 map bridge 31370054065
```

[**no**] **x25 map cmns** *NSAP-prefix* *MAC-address*

[**no**] **x25 map cmns** *NSAP-prefix* [*X.121-address*]

Maps NSAP addresses to either MAC-layer addresses or X.121 addresses. Required after enabling CMNS on a non-serial interface or specifying X.25 encapsulation on a serial interface). The **no** form of the command with address arguments cancels the mapping.

NSAP-prefix—NSAP prefix to map.

MAC-address—MAC address to map.

X.121-address—X.121 address to map.

Example:

```
x25 map cmns 38.8261.17 0800.4e02.1f9f
```

[no] x25 map compressedtcp *ip-address x.121-address [options]*

Specifies a network-protocol-to-X.121 address mapping, such as Internet-to-X.121 or DECnet-to-X.121. This command is required to make the X.25 calls complete for compressed packets. The **no** form of the command disables header compression for the link.

ip-address—IP address.

x.121-address—X.121 address.

options—See options for the preceding **x25 map** command.

Example:

```
x25 map compressed 131.108.9.2 31370054065
```

x25 modulo *modulus*

Sets the modulus. The value of the modulo parameter must agree with that of the device on the other end of the X.25 link. Default: 8.

modulus—8 or 128.

Example:

```
x25 modulo 128
```

x25 nvc *count*

Specifies the maximum number of switched virtual circuits that can be open to one host simultaneously. Default: 1.

count—Circuit count from 1 to 8.

Example:

```
x25 nvc 3
```

x25 ops bytes

Sets the router output packet size to match those of the network. Default: 128 bytes.

byte—Byte count from 128 to 1024. Larger packet sizes are better, because smaller packets require more overhead processing. Use the same byte value for **x25 ips** and **x25 ops** unless your network supports asymmetry between input and output packets.

Example:

```
x25 ops 1024
```

[no] x25 pvc circuit protocol-keyword protocol-address

Establishes or deletes Permanent Virtual Circuits (PVCs). You must specify the required network-protocol-to-X.121 address mapping with an **x25 map** subcommand before you can set up a PVC.

circuit—Virtual circuit channel number. Must be lower than the lower limit of the incoming call range in the virtual circuit channel sequence (set using the **lic** keyword).

protocol-keyword—Protocol type: **ip, decnet, chaos, xns, novell, appletalk, vines, apollo, pup, clns, bridge, or cmns.**

protocol-address—Protocol address of the host at the other end of the PVC.

Example:

```
x25 pvc 2 xns 0123.4567
```

x25 pvc pvc-number interface interface-name pvc-number

Configures a PVC for a given interface.

pvc-number—PVC number that will be used on the local interface (as defined by the primary interface command).

interface-name—Interface type and unit number, as specified by the **interface** keywords.

Example:

```
x25 pvc 2 serial 0 2
```

[no] x25 rpoa *name number*

Specifies a list of transit Recognized Private Operating Agencies (RPOAs) to use, referenced by name.

[no] x25 suppress-called-address

Omits the called (destination) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the called address field. The **no** form of the command resets the default: sending the calling address.

[no] x25 suppress-calling-address

Omits the calling (source) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the calling address field. The **no** form of the command resets the default: sending the calling address.

x25 t10 *seconds*

Sets the limit for the Restart Request retransmission timer (T10) on DCE devices. Default: 60 seconds.

seconds—Number of seconds.

Example:

```
x25 t10 90
```

x25 t11 *seconds*

Sets the limit for the Call Request retransmission timer (T11) on DCE devices. Default: 180 seconds.

seconds—Number of seconds.

Example:

```
x25 t11 90
```

x25 t12 *seconds*

Sets the limit for the Reset Request retransmission timer (T12) on DCE devices. Default: 60 seconds.

seconds—Number of seconds.

Example:

```
x25 t12 90
```

x25 t13 *seconds*

Sets the limit for the Clear Request retransmission timer (T13) on DCE devices. Default: 60 seconds.

seconds—Number of seconds.

Example:

```
x25 t13 90
```

x25 t20 *seconds*

Sets the limit for the Restart Request retransmission timer (T20) on DTE devices. Default: 180 seconds.

seconds—Number of seconds.

Example:

```
x25 t20 90
```

x25 t21 *seconds*

Sets the limit for the Call Request retransmission timer (T21) on DTE devices. Default: 200 seconds.

seconds—Number of seconds.

Example:

```
x25 t21 220
```

x25 t22 *seconds*

Sets the limit for the Reset Request retransmission timer (T22) on DTE devices. Default: 180 seconds.

seconds—Number of seconds.

Example:

```
x25 t22 100
```

x25 t23 *seconds*

Sets the limit for the Clear Request retransmission timer (T23) on DTE devices. Default: 180 seconds.

seconds—Number of seconds.

Example:

```
x25 t23 200
```

x25 th *delay*

Instructs the router to send acknowledgment packets when it is not busy sending other packets, even if the number of input packets has not reached the **win** count. The router sends acknowledgment packets when the number of input packets reaches the count you specify, providing there are no other packets to send. This command improves line responsiveness at the expense of bandwidth. Default: zero.

delay—Number of input packets, from zero to the input window size. The value 1 instructs the router to send an acknowledgment for each packet.

Example:

```
x25 th 1
```

[no] x25 use-source-address

Updates the source address of outgoing calls forwarded over a specific interface. The **no** form of the command prevents the update.

x25 {win|wout} packets

Sets the upper limits on the number of outstanding unacknowledged packets. Set **win** and **wout** to the same value unless your network supports asymmetry between input and output window sizes. Default: 2.

win—Use the keyword **win** to specify the upper limits of the number of outstanding unacknowledged packets in the input window.

wout—Use the keyword **wout** to specify the upper limits of the number of outstanding unacknowledged packets in the output window.

packets—Packets count. For **win**, defines how many packets the router can receive before sending an X.25 acknowledgment. For **wout**, defines how many sent packets can remain unacknowledged before the router uses a hold queue. The value can range from one to the window modulus.

Example:

```
x25 win 3  
x25 wout 3
```

Protocol Configuration

Apollo Domain Global Configuration Commands

[no] apollo maximum-paths *paths*

Sets or cancels the maximum number of multiple paths that the router remembers and uses.

paths—Number of paths to be assigned.

Example:

```
apollo maximum-paths 5
```

[no] apollo route *network network.address*

Specifies static routes for an Apollo Domain network.

network—Network number.

network.address—Address to which to forward packets.

Example:

```
apollo route 45 17.32
```

[no] apollo routing *address*

Enables or disables Apollo routing and specifies which system-wide host address to use.

address—Unique, five-digit hexadecimal host address.

Example:

```
apollo routing 1293c
```

Apollo Domain Interface Subcommands

[no] apollo network *number*

Assigns Apollo Domain network numbers to the appropriate interfaces.

number—Network number, expressed in hexadecimal form.

Example:

```
apollo network 4e
```

apollo update-time *seconds*

Sets the Apollo Domain routing update timers. Default: 30 seconds.

seconds—Number of seconds between updates. The minimum value is 10 seconds.

Example:

```
apollo update-time 20
```

Apollo Domain Access List Commands

[no] apollo access-group *name*

Interface subcommand that specifies the interface on which an Apollo Domain access list is defined.

name—User-defined name for the access list defined by the **apollo access-list** global configuration command.

Example:

```
apollo access-group eng
```

[no] apollo access-group *name*

Interface subcommand that specifies the interface on which an Apollo Domain access list is defined.

name—User-defined name for the access list defined by the **apollo**

access-list global configuration command.

Example:

```
apollo access-group eng
```

[no] apollo access-list *name* {**permit** | **deny**} [*firstnet*]*lastnet.host*
[*wildcard-mask*]

Global configuration command that specifies Apollo Domain access conditions.

name—Defined by the network administrator for the access list.

firstnet- and *lastnet.host*—First and last numbers in a network range. Use the argument *lastnet.host* to specify a single network.

permit or **deny**—Specifies permit or deny condition for this list.

wildcard-mask—Wildcard mask that uses the one bits to ignore the host part of the network address. Host bits corresponding to wildcard mask bits set to zero are used in comparisons.

Example:

```
apollo access-list Engineering permit 89.5
```

AppleTalk Global Configuration Commands

no appletalk arp

Resets the **arp interval** and **arp retransmit** commands to their default values.

appletalk arp interval *milliseconds*

Specifies the interval between retransmission of ARP packets. Default: 33 milliseconds.

milliseconds—Number of milliseconds in the interval. Must be 33 or greater.

Example:

```
appletalk arp interval 100
```

appletalk arp {probe|request} retransmit-count *count*

Specifies the number of times the router will retransmit an ARP packet before abandoning address negotiations and using the selected address. Default: 20.

probe—Sets parameters associated with the router's dynamic address assignment.

request—Sets parameters that the router uses to resolve another node's address.

count—Transmission count. The minimum value is 1.

Example:

```
appletalk arp retransmit-count 25
```

[no] appletalk checksum

Enables or disables the generation and verification of checksums for all AppleTalk packets except routed packets. Default: enabled.

[no] appletalk event-logging

Logs significant events using the logger facility. Logged events include routing changes, zone creation, port status, and address.

[no] appletalk lookup-type *service-type*

Specifies services listed in the **show appletalk nbp** and **show appletalk name-cache EXEC** command display. The **no** form of the command with arguments removes the specified service type from the name cache. The **no** form without arguments removes all names except those relating to Cisco routers.

service-type—Type of AppleTalk service.

[no] appletalk macip dynamic *ip-address* [*ip-address*] **zone** *server-zone*

Allocates a single IP address or a range of IP addresses for assignment to *dynamic* MacIP clients by the MacIP server. The **no** form of the command

with arguments removes the specified dynamic address assignment statement from the configuration; without arguments, it shuts down all running MacIP services.

ip-address—The IP address. To specify a range, supply the first and last IP addresses, separated by a space.

server-zone—The name of the zone containing the MacIP server.

Example:

```
appletalk macip dynamic 131.108.1.28 131.108.1.44 zone
Engineering
```

[no] appletalk macip server *ip-address zone server-zone*

Establishes a new MacIP server. The **no** form of the command with arguments removes a server statement from the configuration; without arguments, it shuts down all running MacIP services.

ip-address—The IP address. You can optionally specify a range by supplying the first and the last IP address.

server-zone—The name of the zone containing the MacIP server.

Example:

```
appletalk macip server 131.108.1.27 zone Engineering
```

[no] appletalk macip static *ip-address [ip-address] zone server-zone*

Defines a range of addresses to be made available to MacIP clients that have reserved invariant IP addresses. The **no** form of the command with arguments removes the specified static address assignment statement from the configuration; without arguments, it shuts down all running MacIP services.

ip-address—The IP address. You can optionally specify a range by supplying the first and the last IP address.

server-zone—The name of the zone containing the MacIP server.

Examples:

```
appletalk macip static 131.108.1.50 131.108.1.66 zone
Engineering
appletalk macip static 131.108.1.81 zone Engineering
```

[no] appletalk name-lookup-interval *seconds*

Sets or cancels the interval between service polls by the router on its AppleTalk interfaces. The value zero or the **no** form of the command disable name lookup. Default: zero.

seconds—Number of seconds.

Example:

```
appletalk name-lookup-interval 1200
```

[no] appletalk permit-partial-zones

Permits partial zones. Even though a network of a zone is denied, the zone is permitted. Default: disabled.

[no] appletalk proxy-npb *network-number zonename*

Assigns a proxy. Required for each zone that has a nonextended-only AppleTalk router connected to a network in the zone.

network-number—Unique network number that the router will advertise as if it were a real network number.

zonename—Name of the zone requiring compatibility support. Only one proxy is needed to support a zone, but additional proxies can be defined with different network numbers, if redundancy is desired.

Example:

```
apple proxy-npb 60 Twilight
```

[no] appletalk require-route-zones

Prevents fake routes (possibly generated by a broken router or corrupt packet) from causing ZIP protocol storms. Requires the router to know the zone name for a route before including it in an update. Default: enabled.

[no] appletalk routing

Enables or disables AppleTalk protocol processing.

[no] appletalk strict-rtmp

Enforces maximum checking of routing packets to ensure their validity.

[no] appletalk timers *update-interval valid-interval invalid-interval*

Changes the intervals used in AppleTalk routing.

update-interval—Number of seconds between routing updates sent to other routers on the network. Default: 10 seconds.

valid-interval—Number of seconds that the router will consider a route valid without having heard a routing update for that route. The value is normally twice the *update-interval*. Default: 20 seconds.

invalid-interval—Number of seconds that the router will wait before marking a route invalid; the default is three times the *valid-interval*, or 60 seconds.

Example:

```
appletalk timers 20 40 120
```

AppleTalk Interface Subcommands

[no] appletalk address *address*

Assigns AppleTalk addresses on the interfaces that will be used for the AppleTalk protocol. Used to configure nonextended interfaces.

address—AppleTalk address.

Example:

```
appletalk address 1.129
```

[no] appletalk cable-range *start-end* [*network.node*]

Designates an interface as being on an extended AppleTalk network.

start-end—First and last network in the range expressed as decimal numbers between 1 and 65279, inclusive. The arguments *start* and *end* can be assigned the same number.

network.node—Network and node number that the router should first use to select the AppleTalk address for this interface.

Example:

```
appletalk cable-range 2-2
```

[no] appletalk discovery

Resets the discovery mode and allows a new cable range to be discovered.

appletalk iptalk *net.node zone*

Encapsulates AppleTalk in IP packets in a manner compatible with the Columbia AppleTalk Package (CAP) IPtalk and the Kinetics IPtalk (KIP) implementation.

net.node—Network node number.

zone—AppleTalk zone name.

Example:

```
appletalk iptalk 30.0 UDPzone
```

appletalk iptalk-baseport *port-number*

Specifies the UDP port number, which is the beginning of the range of UDP ports used in mapping AppleTalk well known DDP socket numbers to UDP ports.

port-number—UDP port number.

Example:

```
appletalk iptalk-baseport 200
```

[no] appletalk send-rtmp

Allows a router to be placed on a network with AppleTalk, enabled but not seen. This allows disabling of routing updates.

[no] appletalk zone *zonename*

Sets the zone name for the connected AppleTalk network. Must be specified after the **appletalk address** or **appletalk cable-range** command if discovery is not enabled. This command may be issued multiple times if it follows the **appletalk cable-range** command.

zonename—Specifies the zone name for the connected AppleTalk network.

Example:

```
appletalk zone twilight
```

AppleTalk Access and Distribution List Commands

access-list *list* {permit**|**deny**} **additional-zones****

Global configuration command that establishes an ACL used for zone-related checks to specify the default action for zones that were not enumerated. Default: deny additional zones.

list—An integer between 600 and 699.

permit or **deny**—Specifies the permit or deny condition.

Example:

```
access-list 600 permit additional zones
```

[no] access-list *list* {permit**|**deny**} **cable-range** *start-end***

Global configuration command that establishes ACL for an extended network. Affects extended networks with starting and ending numbers exactly matching those specified in the command.

list—An integer between 600 and 699.

permit or **deny**—Specifies the permit or deny condition.

start-end—The cable range.

Example:

```
access-list 600 permit cable-range 1000-1010
```

[no] access-list list {permit|deny} includes start-end

Global configuration command that establishes ACL for any network, extended or nonextended, which overlaps any part of the specified range

list—An integer between 600 and 699.

permit or **deny**—Specifies the permit or deny condition.

start-end—The cable range.

Example:

```
access-list 600 permit includes 1000-1010
```

[no] access-list list {permit|deny} network network

Global configuration command that establishes an AppleTalk access control list (ACL) for a single network number. Affects matching nonextended networks and extended networks with the same starting and ending number.

list—An integer between 600 and 699.

permit or **deny**—Specifies the permit or deny condition.

network—AppleTalk network number.

Example:

```
access-list 600 permit network 21
```

[no] access-list list {permit|deny} other-access

Global configuration command that establishes an ACL used as the default for any case not enumerated. Default: deny other access.

[no] access-list *list* {**permit**|**deny**} **within** *start-end*

Global configuration command that establishes an ACL for any network, extended or nonextended, bounded by the specified range. The *start* and *end* numbers are considered to be within the range.

list—An integer between 600 and 699.

permit or **deny**—Specifies the permit or deny condition.

start-end—The cable range.

Example:

```
access-list 600 permit within 1000-1010
```

[no] access-list *list* {**permit**|**deny**} **zone** *zone*

Global configuration command that establishes an ACL that applies to any network that has the specified *zone* in its zone list.

list—An integer between 600 and 699.

permit or **deny**—Specifies the permit or deny condition.

zone—AppleTalk zone name.

Example:

```
access-list 600 deny zone Twilight
```

[no] appletalk access-group *access-list-number*

Interface subcommand that creates a *packet filter*, which prevents any packets from being sent out an interface if the destination network has access denied. Once assigned, any packet that fails the **appletalk access-list** command cannot go out on that interface.

access-list-number—Number of an access list, in the range 600 to 699. If an undefined access list is used, the rule defaults to **permit**. If the condition is not handled by the specified access list, the rule defaults to **deny**, unless permitted via the **other-access** option of the **access-list** global configuration command.

Example:

```
appletalk access-group 699
```

appletalk distribute-list *access-list-number* in

Interface subcommand that filters input from the networks so that AppleTalk network numbers specified by the *access-list-number* argument will not be inserted into the router's AppleTalk routing table when routing updates are received. The **no** form of the command removes this filter.

access-list-number—Number of a predefined access list in the range 600 to 699.

Example:

```
appletalk distribute-list 601 in
```

[no] appletalk distribute-list *access-list-number* out

Interface subcommand that filters routing data generated from zones or networks. The **no** form of the command removes the filter.

access-list-number—Access list number in the range 600 to 699. If an undefined access list is used, the rule defaults to **permit**. If the condition is not handled by the specified access list, the rule defaults to **deny**, unless permitted via the **other-access** option of the **access-list** global configuration command.

Example:

```
appletalk distribute-list 655 out
```

[no] appletalk getzonelist-filter *list*

Interface subcommand that modifies zone-list replies.

list—Access list number in the range 600 to 699. If an undefined access list is used, the rule defaults to **permit**. If the condition is not handled by the specified access list, the rule defaults to **deny**, unless permitted via the **additional-zones** option of the **access-list** global configuration command.

Example:

```
appletalk getzonelist-filter 600
```


CHAOSnet Global Configuration Command

[no] router chaos

Starts or stops the CHAOSnet router process.

DECnet Global Configuration Commands

decnet area-max-cost *value*

Sets the maximum cost specification value for *inter*-area routing.

value—Maximum cost for a route that the router considers usable. The router treats as unreachable any route with a cost greater than the value you specify. The valid range is from 1 to 1022.

Example:

```
decnet area-max-cost 500
```

decnet area-max-hops *value*

Sets the maximum hop count specification value for *inter*-area routing.

value—Maximum number of hops for a route to a distant area that the router may consider usable; the router treats as unreachable any route with a counter greater than the value you specify. The valid range is from 1 to 30.

Example:

```
decnet area-max-hops 21
```

decnet conversion *NSAP-prefix*

Enables conversion.

NSAP-prefix—Value used for the IDP when constructing NSAPs from a Phase IV address.

Example:

```
decnet conversion 47.0004.004d
```

decnet *network-number keywords*

Specifies ATG (Address Translation Gateway).

network-number—Network number, in the range 0 to 3. To configure network 0, do not include the *network-number* modifier.

keywords—One of the DECnet global configuration command keywords; for example, **area-max-cost** or **max-units**.

Examples:

```
decnet 1 node-type area
decnet 0
```

decnet *first-network map virtual-address second-network real-address*

Establishes a translation entry to translate a virtual DECnet address to a real DECnet address for the ATG.

first-network—First DECnet network number, in the range 0 to 3.

map *virtual-address*—Numeric DECnet address that you want to translate to a real address.

second-network—Second DECnet network number, in the range 0 to 3.

real-address—Real DECnet address to which you want to translate the virtual address.

Examples:

```
decnet 0 map 19.5 1 50.1
decnet 0 map 19.6 1 19.1
decnet 1 map 47.1 0 19.1
decnet 1 map 47.2 0 19.3
```

decnet *max-address value*

Determines the largest node number specification allowed in the current area. Default: 255.

value—Node number from 1 to 1023; default is 255.

Example:

```
decnet max-address 300
```

decnet max-area *value*

Determines the largest node number specification allowed in the current area.

value—Largest area number specification the router can handle, from 1 to 63.

Example:

```
decnet max-area 30
```

decnet max-cost *value*

Sets the largest maximum cost specification for intra-area routing.

value—Number, from 1 to 1022.

Example:

```
decnet max-cost 500
```

decnet max-hops *value*

Sets the maximum hop count specification for intra-area routing.

value—Hop count, from 1 to 30.

Example:

```
decnet max-hops 30
```

decnet max-paths *value*

Defines the maximum number of equal cost paths to a destination that the router can maintain. Default: 1 (no multiple paths).

value—Maximum number of equal cost paths, from 1 to 31.

Example:

```
decnet max-paths 2
```

decnet max-visits *value*

Sets the limit on the number of times a packet can pass through a router.

value—Number of times, from 1 to 63.

Example:

```
decnet max-visits 15
```

decnet node-type {**area** | **routing-iv**}

Specifies the node type for the router.

area or **routing-iv**—Specify **area** to make the router exchange traffic directly with routers in other areas, participate in the inter-area (Level 2) routing protocol, and act as an intra-area (Level 1) router for its local area. Specify **routing-iv** to make the router act as an intra-area router and route packets out of the area by taking the least cost path to an inter-area router.

Example:

```
decnet node-type area
```

decnet path-split-mode {**normal** | **interim**}

Sets the mode for splitting the routes between equal cost paths.

normal or **interim**—Specify **normal** to make the router select equal cost paths one after the other. Specify **interim** to make the router route traffic for a single higher level session over the same path. Always specify **interim** if the network is running DECnet VMS Version 4.5 or earlier.

Example:

```
decnet path-split-mode normal
```

[no] decnet router-priority *value*

Sets a priority value for use in determining the default router. The **no** form of the command restores the default: 64.

value—Number between 1 and 127. To configure a router as the designated router, assign it the highest possible node address.

Example:

```
dechnet router-priority 127
```

[no] decnet routing *decnet-address*

Enables or disables DECnet routing for a particular area and node.

decnet-address—An address in DECnet format X.Y, where X is the area number and Y is the node number.

Examples:

```
no decnet routing 6.1023  
dechnet routing 6.10
```

DECnet Interface Subcommands

[no] decnet cost *cost-value*

Sets or removes a cost value for an interface. The **no** form of the command disables DECnet routing for an interface.

cost-value—An integer from 1 to 63. The suggested cost for Ethernet networks is 4. All hosts on the same cable must share the same value.

Example:

```
dechnet cost 4
```

[no] decnet hello-timer *value*

Specifies how often the router sends HELLO messages. The **no** form of the command restores the default of 15 seconds.

value—An interval from 1 to 8191 seconds.

Example:

```
dechnet hello-timer 120
```

[no] decnet route-cache

Enables or disables fast switching and the route cache. The **no** form of the command disables fast switching. Default: enabled.

[no] decnet routing-timer *value*

Specifies how often the router sends routing messages. The **no** form of the command restores the default of 40 seconds.

value—Number of seconds, between 1 and 65,535.

Example:

```
decnet routing-timer 60
```

DECnet Access List Commands

access-list *list* {**permit** | **deny**} *destination destination-mask*

Global configuration command that creates an access list.

list—An integer between 300 and 399 that uniquely identifies the access list.

permit or **deny**—Specifies permit or deny condition for this list.

destination—DECnet address.

destination-mask—The mask, expressed in DECnet address format, with bits set wherever the corresponding bits in the address should be ignored.

Examples:

```
access-list 300 deny 4.51 0.0
access-list 300 permit 2.31 0.
```

[no] access-list *list* {**permit**|**deny**} *source source-mask* [*destination destination-mask*] [*connect-entries*]

Global configuration command that filters *connect initiate* packets by DECnet object type.

list—Access list number in the range 300-399.

permit or **deny**—Specifies permit or deny condition for this list.

source source-mask—Source address and mask.

destination destination-mask—Destination address and mask.

The *connect-entries* argument specifies the entries used to match connect packets. Use the command syntax {**eq**|**neq**} [*src-object*] [*dst-object*] [*identification*]:

eq or **neq**—The keyword **eq** specifies that the item matches the packet if all the specified parts of *src-object*, *dst-object*, and *identification* match data in the packet. The keyword **neq** specifies that the item matches the packet if any of the specified parts do not match the corresponding entry in the packet.

src-object—The keyword **src** and the argument *obj-spec*.

dst-object—The keyword **dst** and the argument *obj-spec*.

identification—The keyword **id**, **password**, or **account**, followed by the argument *regex*.

obj-spec—One of the following: *relop object-number*, **exp** *regex*, **uic** [*group,user*]; in this case, the bracket symbols must be entered.

relop—One of the following relational operator keywords: **eq** (equal to), **neq** (not equal to), **lt** (less than), **gt** (greater than).

object-number—Numeric DECnet object number.

regex—Regular expression that matches a string.

[*group, user*]—Numeric UID expression expressed in decimal, octal, or hexadecimal format.

The argument *obj-spec* is the object specifications, in one of three formats:

- Relational operator and an object number (eq 17).
- Regular expression that matches a string (exp ^SYSTEM\$).
- Matches a UIC (uic [1,4]).

id, password, or account—Type of value that the argument *regex* should match. The special format **eq any** matches any connect packet.

Examples:

```
access-list 300 permit 0.0 63.1023 eq dst eq 27
access-list 300 permit 0.0 63.1023 neq dst eq 17
access-list 300 permit 0.0 63.1023 eq id^SYSTEM$
access-list 300 permit 1.0 0.1023 eq src exp ^SYSTEM$ dst
eq 27
access-list 300 permit 0.0 0.1023 eq any
```

[no] decnet access-group *list*

Interface subcommand that applies or removes an access list.

list—Standard or extended DECnet access list. Standard DECnet access lists apply to destination addresses in this command.

Example:

```
decnet access-group 389
```

[no] decnet in-routing-filter *list*

Provides access control to HELLO messages or routing information received on this interface. The **no** form of the command removes access control.

list—Standard DECnet access list.

Example:

```
decnet in-routing-filter 321
```

[no] decnet out-routing-filter *list*

Applies access control to routing information being sent out on this interface. The **no** form of the command removes access control.

list—Standard DECnet access list.

Example:

```
decnet out-routing-filter 351
```


IP Global Configuration Commands

[no] arp *internet-address hardware-address type [alias]*

Installs a permanent entry in the ARP cache. The router uses this entry to translate 32-bit Internet Protocol addresses into 48-bit hardware addresses.

internet-address—The Internet address that corresponds to the local data-link address specified by the argument *hardware-address*.

hardware-address—Local data-link address.

type—Encapsulation description: **arpa** for Ethernet interfaces, **snap** for FDDI and Token Ring interfaces, or **ultra** for UltraNet interfaces.

alias—Supply this keyword if the router should respond to ARP requests as if it were the owner of the specified IP address.

Example:

```
arp 192.31.7.19 0800.0900.1834 arpa
```

[no] async-bootp *tag [:hostname] data*

Specifies extended BootP requests defined in RFC1084 when the router is configured for SLIP. If no extended BootP commands are entered, by default the software generates a gateway and subnet mask appropriate for the local network.

The argument *tag* is the item being requested, and is one of the following expressed as file name, integer, or IP dotted decimal address:

bootfile—Specifies use of a server boot file from which to download the boot program. Use the optional *:hostname* and *data* arguments to specify the file name.

subnet-mask *mask*—Dotted decimal address specifying the network and local subnetwork mask (as defined by RFC950).

time-offset *offset*—A signed 32-bit integer specifying the time offset of the local subnetwork in seconds from Coordinated Universal Time (UTC).

gateway *address*—Dotted decimal address specifying the IP addresses of

gateways for this subnetwork. A preferred gateway should be listed first.

time-server *address*—Dotted decimal address specifying the IP address of time servers (as defined by RFC868).

IEN116-server *address*—Dotted decimal address specifying the IP address of name servers (as defined by IEN 116).

DNS-server *address*—Dotted decimal address specifying the IP address of Domain Name Servers (as defined by RFC1034).

log-server *address*—Dotted decimal address specifying the IP address of an MIT-LCS UDP log server.

quote-server *address*—Dotted decimal address specifying the IP address of Quote of the Day servers (as defined in RFC865).

lpr-server *address*—Dotted decimal address specifying the IP address of Berkeley UNIX Version 4 BSD servers.

impress-server *address*—Dotted decimal address specifying the IP address of Impress network image servers.

rlp-server *address*—Dotted decimal address specifying the IP address of Resource Location Protocol (RLP) servers (as defined in RFC887).

hostname *name*—The name of the client, (which may or may not be domain qualified, depending upon the site).

bootfile-size *value*—A two-octet value specifying the number of 512 octet (byte) blocks in the default boot file.

:hostname—Host to which this entry applies, expressed as either an IP address or logical host name.

data—List of IP addresses entered in dotted decimal notation or as logical host names, a number, or a quoted string.

Examples:

```
async-bootp bootfile :128.128.1.1 "pcboot"  
async-bootp bootfile :mac "macboot"  
async-bootp subnet-mask 255.255.0.0
```

[no] ip accounting-list *ip-address mask*

Specifies a set of filters to control accounting information for hosts. The **no** form of the command removes this filter.

ip-address—IP address for the host.

mask—Mask with which the source and destination address of each IP datagram is logically ANDed to determine if there is a match.

Example:

```
ip accounting-list 192.31.7.18 255.255.0.0
```

[no] ip accounting-threshold *threshold*

Sets the maximum number of accounting entries to be created. The **no** form of the command removes this limit.

threshold—Maximum number of accounting entries that can be created.

Example:

```
ip accounting-threshold 500
```

[no] ip accounting-transits *count*

Controls the number of transit records that will be stored in the IP accounting database. Transit entries are those that do not match any of the filters specified by **ip-accounting-list** commands.

count—Maximum number of transit records.

Example:

```
ip accounting-transits 100
```

[no] ip domain-list *name*

Defines a list of default domain names to complete unqualified host names. The **no** form of the command deletes a domain name from the list.

name—Domain name to add to or delete from the list.

Example:

```
ip domain-list cisco.com
```

[no] ip domain-lookup

Enables or disables IP Domain Name System-based host-name-to-address

translation. The **no** form of the command disables the feature. Default: enabled.

[no] ip domain-name *name*

Defines the default domain name, which is specified by the argument *name*. The router uses the default domain name to complete names without a dotted domain name. The **no** form of the command deletes the default domain name.

name—Default domain name.

Example:

```
ip domain-name cisco.com
```

[no] ip forward-protocol {**udp** | **nd**} [*port*]

Specifies which protocols and ports are forwarded for an interface with an **ip helper-address**. The **no** form of the command disables forwarding of the specified protocol.

udp or **nd**—Specify **udp** for UDP protocol or **nd** for the ND protocol used by older diskless SUN workstations.

port—If you specified UDP protocol, optionally specify a UDP destination port to control which UDP services are forwarded.

Example:

```
ip forward-protocol udp
```

[no] ip forward-protocol spanning-tree

Permits IP broadcasts to be flooded throughout the internetwork in a controlled fashion. The **no** form of the command prevents flooding.

[no] ip host *name* [*TCP-port-number*] *address1* | [*address2*...*address8*]

Defines a static host-name-to-address mapping in the host cache.

name—Host name.

TCP-port-number—TCP port number. Default: 23 (Telnet)

address—IP address associated with the host name. Up to eight addresses can be supplied.

Example:

```
ip host croff 192.31.7.18
```

[no] ip hp-host *hostname ip-address*

Enables or disables the use of the proxy service.

hostname—Name of the Hewlett-Packard host in the host table.

ip-address—IP address of the host.

Example:

```
ip hp-host bl4zip 131.24.6.27
```

[no] ip ipname-lookup

Specifies or removes the IP IEN-116 Name Server host-name-to-address translation. Default: enabled.

[no] ip name-server *server-address 1 [server-address 2...server-address 6]*

Specifies the addresses of the name servers to use for name and address resolution. Default: all-ones broadcast address (255.255.255.255).

server-address—Internet addresses of up to six name servers in dotted decimal format.

Example:

```
ip name-server 131.108.1.111 131.108.1.2
```

[no] ip routing

Enables or disables IP routing. If the system has optional bridging-enabled software, use the **no** form of the command to set up a system to bridge (not route) IP datagrams. Default: enabled.

[no] ip source-route

Controls the handling of IP datagrams with source routing header options. The **no** form of the command instructs the system to discard IP datagrams containing a source-route option. Default: enabled.

[no] ip subnet-zero

Enables or disables the ability to configure and route to “subnet zero” subnets. Default: disabled.

IP Interface Subcommands

[no] arp {arpa | probe | snap}

Controls the interface-specific handling of IP address resolution into 48-bit Ethernet, FDDI, and Token Ring hardware addresses. Default: **arpa**.

arpa—Specifies standard Ethernet style ARP (RFC 826).

probe—Specifies the HP Probe for IEEE-802.3 networks.

snap—Specifies ARP packets conforming to RFC 1042.

Example:

```
arp probe
```

[no] arp timeout *seconds*

Sets the number of seconds an ARP cache entry will stay in the cache. The **no** form of the command restores the default: 14,400 seconds (4 hours).

seconds—Number of seconds used to age an ARP cache entry for this interface.

Example:

```
arp timeout 7200
```

[no] ip accounting

Enables or disables IP accounting on an interface.

[no] ip address *address net-mask* [**secondary**]

Sets an IP address for an interface.

address—IP address.

net-mask—Subnet mask for the associated network. The subnet mask must be the same for all interfaces connected to subnets of the same network.

secondary—Include this keyword for all IP addresses other than the first.

Examples:

```
ip address 131.108.1.27 255.255.255.0
ip address 192.31.7.17 255.255.255.0 secondary
```

[no] ip broadcast-address *address*

Defines a broadcast address. If you use the **no** form of the command or do not specify a broadcast address, the system uses the default: all ones (255.255.255.255).

address—IP broadcast address for the network.

Example:

```
ip broadcast-address 121.24.43.2
```

[no] ip directed-broadcast

Enables or disables forwarding of directed broadcasts on the interface. Default: enabled.

[no] ip helper-address *address*

Defines a helper address for a specified address. The helper address defines the selective forwarding of UDP broadcasts received on the interface. The **no** form of the command deletes the helper address.

address—Destination broadcast or host address that the router should use when forwarding datagrams.

Example:

```
ip helper-address 128.24.17.111
```

[no] ip mask-reply

Sets the interface to send ICMP Mask Reply messages. Default: disabled.

[no] ip mtu *bytes*

Sets the maximum transmission unit (MTU) or size of IP packets sent on an interface. The **no** form of the command restores the default, which depends on the interface medium.

bytes—Maximum packet size for this interface, from 128 to one less than the maximum for the interface.

Example:

```
ip mtu 300
```

[no] ip probe proxy

Enables or disables HP Probe support, which allows a router to respond to HP Probe Proxy Name requests. Default: disabled.

[no] ip proxy arp

Enables or disables proxy ARP on the interface. Default: enabled.

[no] ip redirects

Enables or disables the sending of ICMP redirects on this interface.
Default: enabled.

[no] ip route-cache [cbus]

Controls the use of outgoing packets on a high-speed switching cache for IP routing. The cache is enabled by default and allows load-balancing for individual destinations; autonomous switching is disabled by default. The **no** form of the command disables fast-switching, enabling load-balancing on a per-packet basis.

cbus—Enables autonomous switching.

no ip security

Restores an interface to its default state: dedicated, unclassified Genser, with no extended state allowed.

[no] ip security add

Adds a basic security option to all datagrams leaving the router on the specified interface. The **no** form of the command disables this function.

ip security dedicated level authority [authority..]

Sets or removes the requested level of classification and authority on the interface.

level—Degree of sensitivity of information: **reserved4, TopSecret, Secret, Confidential, Reserved3, Reserved2, Unclassified, Reserved1.**

authority—Organization that defines the set of security levels that will be used in a network: **Genser, Sio-Esi, SCI, or NSA.**

Example:

```
ip security dedicated confidential Genser
```

[no] ip security extended-allowed

Allows or rejects datagrams with an extended security option on the specified interface.

[no] ip security first

Prioritizes the presence of security options on a datagram.

[no] ip security ignore-authorities

Sets an interface to ignore the authority fields of all incoming datagrams. The **no** form of the command removes the setting.

[no] ip security implicit-labelling [*level authority [authority [authority...]]*]

Sets the interface to accept datagrams, even if the packets do not include a security option. The **no** form of the command removes the setting.

level—Degree of sensitivity of information: **reserved4, TopSecret, Secret, Confidential, Reserved3, Reserved2, Unclassified, Reserved1.**

authority—Organization that defines the set of security levels that will be used in a network: **Genser, Sio-Esi, SCI, or NSA.**

Example:

```
ip security implicit-labelling confidential Genser
```

ip security multilevel *level1 [authority...]* **to** *level2 authority2 [authority2...]*

Sets the requested range of classification and authority on the interface. Traffic entering or leaving the system must have a security option that belongs in the specified range. The **no** form of the command removes the setting.

level—Degree of sensitivity: **reserved4, TopSecret, Secret, Confidential, Reserved3, Reserved2, Unclassified, Reserved1.**

authority—Organization that defines the set of security levels that will be used in a network: **Genser**, **Sio-Esi**, **SCI**, or **NSA**.

Example:

```
ip security multilevel Confidential Genser to TopSecret
Genser
```

[no] ip security strip

Removes any basic security option on all datagrams leaving the router on the specified interface. The **no** form of the command disables the function.

[no] ip split-horizon

Enables or disables the split horizon mechanism. The default for interfaces without frame relay or SMDS encapsulation is enabled. The default for all other interfaces is disabled.

[no] ip tcp compression-connections *number*

Sets the maximum number of connections per interface that the compression cache can support. Default: 16.

number—Number of connections, from 3 to 256

Example:

```
ip tcp compression-connections 256
```

[no] ip tcp header-compression [passive]

Enables TCP header compression. The **no** form of the command disables the compression. Default: disabled.

passive—Sets the interface to compress outgoing traffic on the interface for a specific destination, but only if incoming traffic is compressed.

Example:

```
ip tcp header-compression passive
```

[no] ip unnumbered *interface-name*

Enables or disables IP processing on a serial interface, but does not assign an explicit IP address to the interface.

interface-name—Name of another interface on which the router has assigned an IP address. Do not use this interface or another unnumbered interface.

Example:

```
ip unnumbered ethernet 0
```

[no] ip unreachable

Enables or disables the ability to send ICMP unreachable messages on an interface. Default: enabled.

transmit-interface *interface-name*

Assigns a transmit interface to a receive-only interface.

interface-name—Name of the interface to which you want to convert the interface designated as the source of the route.

Example:

```
transmit-interface Ethernet 0
```

Serial Line IP (SLIP) Line Subcommands

no slip

Cancels SLIP support on the line.

slip address *internet-address*

Specifies the Internet address assigned to the SLIP client at the other end of the serial line connection.

internet-address—Internet address. Must be on the same network or subnet as the router's network interface.

Example:

```
slip address 128.73.98.2
```

slip address dynamic [*IP-address*]

When issued without an IP address, allows the IP address associated with a SLIP line to be assigned upon access. This feature is supported when a TACACS server is used. When issued with an IP address (*IP-address*), allows a default address to be specified upon access.

Example:

```
slip address dynamic 124.201.14.3
```

slip dedicated

Places the line in SLIP mode permanently. The router does not create an EXEC on this line, so the line is not available for normal interactive use.

slip hold-queue *packets*

Specifies the limit of the SLIP output queue, which stores packets received from the network waiting to be sent to the SLIP client. Default: 2.

packet—Maximum number of packets.

Example:

```
slip hold-queue 4
```

slip interactive

Allows the line to be used in either SLIP mode or interactive mode. Interactive mode is restored when the modem is disconnected or the line cleared.

slip mtu *bytes*

Specifies the size of the largest Internet packet that the SLIP support can handle. Default: 1500 bytes.

byte—Maximum number of bytes.

Example:

```
slip mtu 3000
```

speed *baud*

Sets the transmit and receive speeds for the line.

baud—100, 1200, 2400, 4800, 9600, 19200, or 38400.

Example:

```
speed 9600
```

IP and SLIP Access List Commands

[no] access-class *list* {**in**|**out**}

Line subcommand that restricts incoming and outgoing connections between a particular virtual terminal line and the addresses in an access list.

list—An integer from 1 to 99 that identifies a specific access list of Internet addresses.

in or **out**—Use the keyword **in** to restrict incoming connections; use the keyword **out** to restrict outgoing Telnet connections.

Example:

```
access class 23 in
```

[no] access-list *list* {**permit**|**deny**} *address wildcard-mask*

Global configuration command that creates or removes an access list.

list—An IP list number from 1 to 99.

permit or **deny**—Specifies permit or deny condition for this list.

address—32-bit, dotted decimal notation IP address to which the router compares the address being tested.

wildcard-mask—Wildcard mask bits for the address in 32-bit, dotted decimal notation.

Example:

```
access-list 1 permit 192.5.34.0 0.0.0.255
```

[no] access-list *list* { **permit** | **deny** } *protocol source source-mask destination destination-mask [operator operand] [established]*

Global configuration command that creates or removes an extended access list.

list—An IP list number from 100 to 199.

permit or **deny**—Specifies permit or deny condition for this list.

protocol—One of the supported protocol keywords: **ip**, **tcp**, **udp**, or **icmp**.

source—An IP address in 32-bit, dotted decimal notation.

source-mask—Mask bits for the source address in 32-bit, dotted decimal notation.

destination—Destination address in 32-bit, dotted decimal notation.

destination-mask—Destination address mask in 32-bit, dotted decimal notation.

operator—If **tcp** or **udp** is the protocol, use these optional arguments to compare destination ports, service access points, or contact names. Specify **lt** (less than), **gt** (greater than), **eq** (equal to), or **neq** (not equal).

operand—If **tcp** or **udp** is the protocol, specify the decimal destination port for the specified protocol.

established—For the TCP protocol, use the **established** keyword to match TCP datagrams that have the ACK or RST bits set.

Example:

```
access-list 189 permit tcp 128.88.0.0 0.0.255.255 0.0.0.0
255.255.255.255
access-list 190 permit tcp 0.0.0.0 255.255.255.255
128.88.1.2 eq 25
```

[no] ip access-group *list*

Interface subcommand that defines an access group.

list—Standard or extended IP access list number from 1 to 199.

Example:

```
access-group 101
```

slip access-class *number* {**in** | **out**}

Line subcommand that configures an access list to be used on packets to or from the SLIP host.

number—IP access list number.

in—Configures list for packets from the SLIP host

out—Transmits only those packets whose IP source address appears on the access list.

Example:

```
slip access-class 3 in
```

IP Routing Protocols Global Configuration Commands

[no] autonomous-system *local-AS*

Specifies an autonomous system (AS) number. The **no** form of the command removes the AS number.

local-AS—Local autonomous system number to which the router belongs.

Example:

```
autonomous-system 109
```

[no] ip default-network *network-number*

Instructs a smart router to generate dynamic default information and pass it to other routers. The **no** form of the command removes the instruction.

network-number—Network number.

Example:

```
ip default-network 128.99.0.0
```

[no] ip route *network address*

Establishes static routes.

network-address—Network address for which you are establishing a static route.

Example:

```
ip route 131.161.7.12
```

[no] router protocol [*autonomous-system*]

Selects the IP routing process. Must be entered before the router sub-commands that further define and tune the routing process.

protocol—Protocol-type keyword: **rip**, **egp**, **hello**, **bgp**, or **igrp**.

autonomous-system—For IGRP, BGP, or IGP protocol only: the number of an autonomous system.

Example:

```
router igrp 120
router hello
```

[no] router egp 0

A specific version of the **router** global configuration command that specifies a core gateway. Allows a specific router to have an EGP process that will enable it to behave like a peer to any reachable autonomous system.

[no] router ospf *ospf-process-id*

Enables OSPF for the router. You can specify multiple OSPF routing processes in each router.

ospf-process-id—An internally used identification parameter, expressed as a positive integer.

Example:

```
router ospf 109
```

IP Routing Protocols Interface Subcommands

ip address *address mask* [**secondary**]

Specifies the IP address on an interface.

address—IP address.

mask—Subnet mask.

secondary—Include the keyword **secondary** to allow multiple IP addresses per interface.

Example:

```
ip address 181.117.17.16 255.255.255.0
```

[no] ip gdp

Enables or disables GDP routing with all default parameters.

[no] ip gdp holdtime *seconds*

Enables or disables GDP routing, keeping priority and reporting interval at their default settings.

seconds—Holdtime, in seconds.

Example:

```
ip gdp holdtime 20
```

[no] ip gdp priority *number*

Enables or disables GDP routing, maintaining report time at 5 seconds for Ethernet networks, and holdtime at 15 seconds.

number—Priority number.

Example:

```
ip gdp priority 95
```

[no] ip gdp reporttime *seconds*

Enables or disables GDP routing, maintaining a priority of 100 and hold time of 15 seconds.

seconds—Report time.

Example:

```
ip gdp reporttime 10
```

[no] ip ospf authentication-key *8-bytes-of-password*

Assigns or cancels a password to be used by neighboring routers on a wire that employs OSPF's simple password authentication.

8-bytes-of-password—String of characters that you can enter from the keyboard, up to eight bytes in length.

Example:

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

[no] ip ospf cost *cost*

Specifies or removes the cost of sending a packet on an interface, for OSPF only.

cost—Link state metric advertised as the link cost in the router's router links advertisement. The software does not support Type of Service (TOS), so you can assign only one cost per interface.

In general, the path cost is calculated as follows:

$$\frac{10^8}{\textit{Bandwidth}}$$

The following table lists the media types supported and their default costs.

Media	Default Cost
56 Kbps Serial Link	1785
64 Kbps Serial Link	1562
T1 (1.544 Mbps Serial Link)	65
E1 (2.048 Mbps Serial Link)	48
4 Mbps Token Ring	25
Ethernet	10
16 Mbps Token Ring	6
FDDI	1

[no] ip ospf dead-interval *number-of-seconds*

Sets or cancels the number of seconds that neighboring routers wait after seeing this router's HELLO packets before declaring the router down, for OSPF only. This value is advertised in the router's HELLO packets in the *DeadInt* field, and must be the same for all routers attached to a common network. Default: four times the HELLO interval.

number of-second—Unsigned integer value.

Example:

```
is ospf dead-interval 60
```

[no] ip ospf hello-interval *number-of-seconds*

Sets or cancels the number of seconds between the HELLO packets that the router sends on the interface, for OSPF only. This value is advertised in the router's HELLO packets, and must be the same for all routers attached to a common network.

number of-second—Unsigned integer value.

Example:

```
is ospf hello-interval 15
```

[no] ip ospf priority *8-bit-number*

Establishes or cancels a Router Priority, which helps determine the Designated Router for a network, for OSPF only. Default: zero.

8-bit-number—8-bit unsigned integer.

Example:

```
ip ospf priority 4
```

[no] ip ospf retransmit-interval *number-of-seconds*

Sets the number of seconds between link state advertisement retransmissions, for adjacencies belonging to the interface. Default: 5 seconds.

number-of-seconds—Number of seconds. Should be greater than the expected round-trip delay between any two routers on the attached network. The value should be larger for serial lines and virtual links.

Example:

```
ip ospf retransmit-interval 7
```

[no] ip ospf transmit-delay *number-of-seconds*

Sets or cancels the estimated number of seconds it takes to transmit a link state update packet on the interface, for OSPF only. The value should include the transmission and propagation delays for the interface. Default: 1 second.

number-of-seconds—Number of seconds; an integer.

Example:

```
ip ospf transmit-delay 2
```

[no] ip route *network address*

Establishes static routes.

network-address—Network address for which you are establishing a static route.

Example:

```
ip route 131.161.7.12
```

IP Routing Protocols Router Subcommands

[no] area *area-id* authentication

Enables authentication for an area, for OSPF only. The authentication type (AuType 0 or AuType1) must be the same for all routers in an area.

area-id—The Area ID of the area for which authentication is to be enabled.

Example:

```
area 12 authentication
```

[no] area *area-id* stub

[no] area *area-id* default-cost *cost*

The two commands define an area as a stub area, for OSPF only. This command is used only on an area border router attached to a stub.

area-id—The Area ID for the stub area, specified as either a decimal value or an IP address.

cost—Cost for the default external route used for the stub area, expressed as a 32-bit number.

Examples:

```
area 12 stub
area 12 default-cost 1001
```

[no] area *area-id* range *address mask*

Advertises a single summary route to other areas, for OSPF only. This command is used only with area border routers.

area-ID—Area ID for the area about which routes are to be summarized, specified as either a decimal value or an IP address.

address—Standard IP address.

mask—Standard IP mask.

Example:

```
area 9 range 131.120.18.0 255.255.0.0
```

[no] area *area-id* **virtual-link** *router-id* [**hello-interval** *number-of-seconds*] **retransmit-interval** *number-of-seconds*] [**transmit-delay** *number-of-seconds*] [**dead-interval** *number-of-seconds*] [**authentication-key** *number-of-seconds*]

Defines virtual links, for OSPF only.

area-id—Area ID assigned to the transit area for the virtual link.

router-id—Router ID associated with the virtual link neighbor.

number-of-seconds—Number of seconds between the HELLO packets that the router sends on the interface, expressed as an unsigned integer.

This value must be the same for all routers attached to a common network. Default: 10 seconds.

[no] default-information allowed {in | out}

Controls the handling of default information between multiple IGRP processes. Default: enabled.

in or **out**—Use the **in** keyword with the **no form** of the command to suppress IGRP exterior or default routes when received by an IGRP process. Use the **out** keyword with the **no form** of the command to suppress IGRP exterior routes in updates.

Example:

```
no default-information allowed out
```

[no] default-information originate

Configures EGP to generate a default route. If the next hop for the default route can be advertised as a third party, it is included as a third party.

[no] default-information originate metric *metric-value* **metric-type** *type-value*

Enables or disables the generation of a default route into an OSPF domain. Must be used with a **redistribute** command.

metric *metric-value*—Specifies the link state cost to be assigned to the default route. The argument *metric-value* is a dimensionless link state cost, formed as a 24-bit decimal.

metric-type *type-value*—Specifies the external link type associated with the default route advertised into the OSPF routing domain. The *type-value* argument can be 1 (Type 1 external route) or 2 (Type 2 external route). Default: 2.

Example:

```
default-information originate metric 100 metric-type 1
```

default-metric *bandwidth delay reliability loading mtu*

Sets metrics for IGRP only.

bandwidth—Minimum bandwidth of the route in kilobits per second.

delay—Route delay in tens of microseconds.

reliability—Likelihood of successful packet transmission, expressed as a number between 0 and 255, where 255 is 100 percent reliability.

loading—Effective bandwidth of the route in kilobits per second.

mtu—Minimum Maximum Transmission Unit (MTU) of the route.

default-metric *number*

Sets metrics for RIP, EGP, BGP, and HELLO, which use scalar, single-valued metrics.

number—Default metric value appropriate for the specified routing protocol, expressed as an unsigned integer.

Example:

```
default-metric 10000
```

no default-metric

Instructs the current routing protocol to resume using the built-in automatic metric translations.

[no] distance *weight* [[*ip-source-address ip-address-mask*] [*access-list-number*]]

Defines or deletes an administrative distance.

weight—Integer from 10 to 255 that specifies a default administrative distance that the router uses when no other specification exists for a routing information source.

ip-source-address—Internet address that identifies a router, network, or subnet to which the weight value applies.

ip-address-mask—A mask that specifies which bits, if any, to ignore in the address value, in dotted-decimal format.

access-list-number—Number of a standard IP access list.

Example:

```
distance 90 192.31.7.0 0.0.0.255
```

[no] distance bgp *external-distance internal-distance local-distance*

Specifies administrative distance.

external-distance—Value for BGP external routes. Default: 20.

internal-distance—Value for BGP internal routes. Default: 200.

local-distance—Value for BGP local routes. Default: 200.

Example:

```
distance bgp 40 100 200
```

[no] distribute-list *access-list-number in* [*interface-name*]

Filters networks received in updates, or cancels the filter.

access-list-number—Standard IP access list number.

in—Include to suppress incoming routing updates.

interface-name—Interface on which the access list should be applied to incoming updates. If no interface is specified, the access list is applied to all incoming updates.

Example:

```
distribute-list 1 in Ethernet 0
```

[no] distribute-list *access-list-number* **out** [*interface-name* | *routing-process*]

Suppresses networks so that they are not sent in updates.

access-list-number—Standard IP access list number.

in—Include to apply the access list to outgoing routing updates.

interface-name or *routing-process*—To restrict the routing updates sent to a specific interface, specify the interface. To redistribute networks, specify the routing process name.

Example:

```
distribute-list 3 out igmp 109
```

[no] metric holddown

For IGRP only. Disables or re-enables holddown. Use this command only if the entire autonomous system (AS) is running Version 8.2(5) or later.

[no] metric maximum-hops *hops*

For IGRP only. Causes the IP routing software to advertise as unreachable those routes with a hop count greater than the assigned value.

hops—Hop count, from 1 to 255. Default: 100.

Example:

```
metric maximum-hops 150
```

[no] metric weights *TOS K1 K2 K3 K4 K5*

Allows the tuning of the IGRP metric calculation for a particular Type of Service (TOS).

TOS—Use the parameter 0.

K1-K5—Constants for the equation that converts an IGRP metric vector into a scalar quantity.

The following table lists the delay and bandwidth values for each supported media type.

Media Type	Delay	Bandwidth
Satellite	200,000 (2 sec)	20 (500 Mbit)
Ethernet	100 (1 ms)	1,000
1.544 Mbps	2000 (20 ms)	6,476
64 Kbps	2000	156,250
56 Kbps	2000	178,571
10 Kbps	2000	1,000,000
1 Kbps	2000	10,000,000

[no] neighbor *address*

Creates a list of neighbor routers. The **no** form of the command removes the entry.

address—IP address of a peer router with which routing information will be exchanged.

Example:

```
neighbor 192.31.7.0
```

[no] neighbor any [*list*]

[no] neighbor any third-party *IP-address* [**internal** | **external**]

Controls how an EGP process determines which routers to treat as peers. Used with the **router egp 0** router subcommand.

list—Access list number. When this argument is supplied, the neighbor is accepted by the access list to be allowed to peer with the EGP process.

third-party *IP-address*—Provides the IP address to be used as the next hop in EGP advertisements.

internal or **external**—Indicates whether the third-party router should be listed in the internal or external section of the EGP update.

Examples:

```
neighbor any 2
neighbor any third-party 10.1.1.1
```

[no] neighbor *address* distribute-list *list* {in | out}

Distributes neighbor information as specified in an address list for BGP.

address—IP address of the neighbor router.

list—Access list to be applied to incoming or outgoing updates.

in or **out**—If the access list is to be applied to incoming updates, use the keyword **in**. If the access list is to be applied to outgoing updates, use the keyword **out**.

Examples:

```
neighbor 192.31.7.0 distribute-list 41 out
neighbor 120.23.4.1 distribute-list 39 in
```

[no] neighbor *address* filter-as *number* deny

Filters neighbor information for an address in a specific autonomous system (AS) for BGP. Instructs the router to ignore the information learned about the network.

address—IP address of the neighbor router.

number—Autonomous system number of the neighbor routers that you want to filter.

Example:

```
neighbor 120.23.4.1 filter-as 20 deny
```

[no] neighbor *address* filter-as *number* permit *weight*

Filters neighbor information for an address in a specific autonomous system (AS) for BGP. Allows incoming information to be added to the routing table.

address—IP address of the neighbor router.

number—Autonomous system number of the neighbor routers that you want to filter.

weight—Weight associated with incoming information that is added to the routing table.

Example:

```
neighbor 120.23.4.1 filter-as 20 permit 60
```

[no] neighbor *address filter-list list {in | out}*

[no] neighbor *address filter-list list weight weight*

Establishes or removes filters. You can specify multiple weight filters for a single neighbor, but only one in or out filter.

address—IP address of the neighbor router.

list—Predefined BGP access list number.

in or **out**—If the access list is to be applied to incoming updates, use the keyword **in**. If the access list is to be applied to outgoing updates, use the keyword **out**.

weight *weight*—Assigns a relative importance to a specific list.

Examples:

```
neighbor 192.31.7.0 filter-list 41 out
neighbor 120.23.4.1 filter-list weight 10
```

[no] neighbor *address [priority 8-bit-number] [poll-interval number-of-seconds] interface type unit-number*

Configures routers interconnecting to non-broadcast networks, for OSPF only. The **no form** of the command removes the neighbor with the specified IP address from the list.

address—Interface IP address of the neighbor.

priority *8-bit-number*—Router Priority value of the the nonbroadcast neighbor associated with the specified IP address.

poll-interval *number-of-seconds*—Specifies the interval at which HELLO packets are sent to a neighboring router that has become inactive. This interval should be much larger than the Hello Interval. Default: 120 seconds.

Example:

```
neighbor 120.23.4.77
```

[no] neighbor *address* **remote-as** *autonomous-system*

Adds a neighbor entry to the routing table for BGP.

address—IP address of the neighbor router.

autonomous-system—Autonomous system number of the neighbor router.

Example:

```
neighbor 131.108.1.2 remote-as 109
```

[no] neighbor *address* **third-party** *third-party-ip-address*
[internal|external]

Adds third-party information to routing updates.

address—IP address of the EGP peer.

third-party-ip-address—Address of the third party on the network shared by the Cisco router and the EGP peer.

internal or **external**—Indicates whether the third-party router should be listed in the internal or external section of the EGP update.

Example:

```
neighbor 131.108.6.5 third-party 131.108.6.99 internal
```

[no] neighbor *IP-address* **version** *value*

Configures the router to handle only Version 2 of the BGP protocol. The **no** form of the command returns the version to the default state for that neighbor.

IP-address—Address of the BGP-speaking neighbor

value—Version number; must be 2.

Example:

```
neighbor 131.104.27.2 version 2
```

[no] neighbor *address weight weight*

Specifies a weight to assign to a neighbor connection. The router chooses as the preferred route the route with the highest weight.

address—Specific neighbor connection to which you want to assign the *weight*.

weight—Weight to assign.

Example:

```
neighbor 131.99.87.2 weight 50
```

[no] network *network-number*

Specifies a list of networks to be advertised as originating within an AS. Or, for EGP, specifies the network to be advertised to the EGP peers of an EGP routing process. The **no** form of the command removes an entry from the list.

network-number—Network number.

Example:

```
network 129.140.0.0
```

[no] network *address wildcard-mask area area-id*

Specifies a range of IP addresses for any area in which OSPF is used as a routing protocol.

address—An IP address.

wildcard-mask—An IP-address-type mask that includes “don’t care” bits.

area-id—An area to be associated with the OSPF address range, specified as either a decimal value or an IP address.

Example:

```
network 131.108.20.0 0.0.0.255 area 10.9.50.0
network 131.108.0.0. 0.0.255.255 area 2
```

[no] network *address* backdoor

Specifies a back-door route to a BGP border router that will provide better information about the network.

address—Address of the network to which you want to set up a back-door route.

Example:

```
network 192.31.7.0 backdoor
```

[no] offset-list *list* {in** | **out**} *offset***

For IGRP, RIP, and HELLO only. Adds or removes a positive offset to incoming and outgoing metrics for networks matching an access list.

list—Access list. Specify zero to apply the offset to all metrics.

in or **out**—Indicate whether the offset applies to incoming (**in**) or outgoing (**out**) metrics.

offset—Offset. For IGRP, the offset is added to the delay component only.

Example:

```
offset-list 121 out 110
```

[no] passive-interface *interface*

Disables or enables the routing of updates on an interface.

interface—Interface name.

Example:

```
passive-interface fddi1
```

[no] redistribute *process-name* [*AS-number*]

Passes routing information among routing protocols.

process-name—Routing information source: **static**, **rip**, **egp**, **hello**, or **igrp**.

AS-number—If you specified **igrp** or **egp**, supply the autonomous system number.

Example:

```
redistribute egp 4
redistribute static
```

[no] redistribute *protocol* [*source-id*] [**metric** *metric-value*] [**metric-type** *type-value*] [**tag** *tag-value*] [**subnets**]

Redistributes routes from other OSPF routing domains and non-OSPF routing domains into a specific OSPF routing domain.

protocol—Source protocol from which routes are being distributed: **bgp**, **egp**, **hello**, **igrp**, **ospf**, **rip**, or **static**.

source-id—Autonomous system (IGRP) or an OSPF process ID from which routes are to be redistributed. Do not include this argument if you used the keyword **hello** or **rip**.

metric *metric-value*—Specifies the link state cost to be assigned to the default route. The argument *metric-value* is a dimensionless link state cost, formed as a 24-bit decimal.

metric-type *type-value*—Specifies the external link type associated with the default route advertised into the OSPF routing domain. The *type-value* argument can be 1 (Type 1 external route) or 2 (Type 2 external route). Default: 2.

tag *tag-value*—32-bit decimal value attached to each external route.

subnets—Specifies the scope of redistribution for the specified protocol.

Example:

```
redistribute hello metric 100 metric-type 1
```

[no] timers basic *update invalid holddown flush*

Adjusts timers. Use the **show ip protocols** command to display defaults and current values. The **no** form of the command restores the default.

update—Rate at which updates are sent.

invalid—Number of seconds after which a route is declared invalid. The value of *invalid* should be three times the value of *update*.

holddown—Number of seconds during which routing information regarding better paths is suppressed. The value of *holddown* should be at least three times the value of *update*.

flush—Number of seconds that must pass before the route is removed from the routing table. The value of *flush* should be equal to or greater than the sum of the values of *invalid* and *holddown*.

Example:

```
timers basic 5 15 15 30
```

[no] timers bgp *keepalive holdtime*

Adjusts BGP timers. The **no** form of the command restores the default.

keepalive—Frequency with which the router sends keepalive messages to its peer. Default: 60 seconds.

holdtime—Number of seconds that the router waits for a keepalive message before declaring a peer dead. Default: 180 seconds.

Example:

```
timers bgp 50 100
```

[no] timers egp *hello poll*

Adjusts EGP timers. The **no** form of the command restores the default.

hello—Interval at which HELLO packets are sent. Default: 60 seconds.

poll—Interval at which polling is performed. Default: 180 seconds.

Example:

```
timers egp 30 90
```

variance *multiplier*

Controls the amount of load balancing that IGRP can perform. Default: 1

multiplier—Nonzero positive integer.

Example:

```
variance 5
```

IP Routing Protocols Access List Commands

[no] ip as-path access-list *list* {**permit** | **deny**} *as-regular-expression*

Global configuration command that defines a BGP-related access list.

list—Access list number.

permit or **deny**—Specifies the permit or deny condition.

as-regular-expression—A regular expression, as explained in the *Router Products Configuration and Reference*, Appendix D, “Pattern Matching.”

Example:

```
ip as-path access-list 4 permit ^net
```

ISO CLNS Global Configuration Commands

[no] clns configuration-time *seconds*

Specifies the rate at which ESH and ISH packets are sent. The **no** form of the command restores the default: 60.

seconds—Number of seconds.

Example:

```
clns configuration-time 45
```

[no] clns holding-time *seconds*

Allows the sender of an ESH or ISH packet to specify the length of time during which the information in the HELLO packets will be believed. The **no** form of the command restores the default: 300 seconds.

seconds—Number of seconds.

Example:

```
clns holding-time 240
```

clns host *name nsap*

Creates a name for a Network Service Access Point (NSAP). The name can be used instead of the long set of numbers associated with an NSAP.

name—Name for the NSAP.

nsap—Numbers associated with the NSAP.

Example:

```
clns host nsap1 47.0004.004d.3132.3334.3536.00
```

[no] clns net *NET-address*

Assigns static network addresses. If a router is configured to support ISO CLNS and is not configured to dynamically route CLNS packets using a ISO-IGRP or IS-IS, you must use this command to assign an address to the router. The **no clns net** command removes any previously configured NET address. Default: none.

NET-address—Network Entity Title (NET) address.

Example:

```
clns net 47.0005.0001.0000.0001.000.00
```

[no] clns packet-lifetime *number*

Specifies the initial lifetime for locally generated packets. Default: 64.

number—Number of seconds.

Example:

```
clns packet-lifetime 32
```

[no] clns routing

Enables or disables routing of CLNS packets.

[no] clns want-er pdu

Specifies whether the router requests error PDUs on packets sourced by the router. Default: enabled.

ISO CLNS Interface Subcommands

[no] clns checksum

Enables or disables checksum generation when ISO CLNS routing software sources a CLNS packet. The **no** form of the command restores the default: disabled.

[no] clns cluster-alias

Configures or cancels DECnet Phase V cluster aliasing. When enabled, the router can advertise the same system identifier as other systems in end-system HELLO messages.

[no] clns congestion-threshold *number*

Sets the congestion experience bit if the output queue contains more than the specified number of packets. The **no** form of the command prevents this bit from being set. Default: 4.

number—Maximum number of packets that are accepted before the router sets the congestion experience bit.

Example:

```
clns congestion-threshold 3
```

[no] clns enable

Enables static or dynamic routing. The **no** form of the command configures the interface to pass ISO CLNS packet traffic to end systems.

[no] clns erpdu-interval *milliseconds*

Determines the minimum interval time, in milliseconds, between ERPDUs. The **no** form of the command turns off the interval rate and sets no limit. Default: 10.

milliseconds—Number of milliseconds between ERPDUs.

Example:

```
clns erpdu-interval 5
```

[no] clns {es-neighbor|is-neighbor} nsap snpa

Lists all End Systems that will be used when mapping information is statically entered.

es-neighbor or **is-neighbor**—Specify **es-neighbor** for End Systems, or specify **is-neighbor** for Intermediate Systems.

nsap—CLNS address.

snpa—MAC addresses.

Example:

```
clns is-neighbor 47.0004.0021.0001.0000.0000.00 3101
```

[no] clns {es-neighbor|is-neighbor} nsap snpa [X.25-facilities-info]

Lists all End Systems or Intermediate Systems that will be used when mapping information is entered statically. The SNPAs are the X.25 network addresses (X.121 addresses). These are usually assigned by the X.25 network provider.

es-neighbor or **is-neighbor**—Specify **es-neighbor** for End Systems, or specify **is-neighbor** for Intermediate Systems.

nsap—CLNS address.

snpa—MAC addresses.

X.25-facilities-info—Specifies nondefault packet and window size, reverse charge information, and so on.

Examples:

```
clns is-neighbor 47.0004.0021.0001.0000.0000.00 3101
window-size 7 7 packet-size 512 512
clns is-neighbor 47.0004.004d.3132.3334.3536.00 310117
```

[no] clns es-neighbor nsap snpa [X.25-facilities-info]

Lists all End Systems that the router will use when mapping information is entered statically.

nsap—CLNS address.

snpa—Data link address.

X.25-facilities-info—Nondefault packet and window size and reverse charge information.

Example:

```
clns es-neighbor 47.0004.004d.3132.3334.3536.00 310117
```

[no] clns mtu *size*

Sets the MTU packet size for the interface. The **no** form of the command restores the default: the maximum packet size for the interface.

size—MTU packet size for the interface. The minimum value is 512.

Example:

```
clns mtu 512
```

[no] clns net *{NSAP-address/name}*

Assigns an NSAP address or node name for a router interface. The **no** form of the command removes any previously configured NSAP-address.

NSAP-address—NSAP address.

name—Node name.

Example:

```
clns net 47.0004.004D.0003.0000.0C00.62E6.00
```

[no] clns rdpdu-interval *milliseconds*

Determines the minimum interval time, in milliseconds, between RDPDUs. The **no** form of the command sets no limit. Default: 100.

milliseconds—Minimum interval time.

Example:

```
clns rdpdu-interval 50
```

[no] clns rdpdu-mask

Enables or disables the address mask on RDPDUs.

[no] clns route-cache

Enables or disables fast switching through the cache. Default: enabled.

[no] clns send-er pdu

Allows CLNS to send an error PDU when it detects an error in a data PDU. The **no** form of the command disables the function.

Default: enabled.

[no] clns send-r pdu

Allows CLNS to send redirect PDUs when a better route for a given host is known. The **no** form of the command disables the function.

Default: enabled.

ISO CLNS Routing Protocols Global Configuration Commands

[no] clns route *nsap-prefix interface-type unit [SNPA-address]*

Creates a static route for a specific interface. The **no** form of the command with the *nsap-prefix* argument cancels the static route.

nsap-prefix—Prefix of NSAPs to be forwarded.

interface-type—Interface type.

unit—Interface number.

SNPA-address—SNPA address to which the NSAPs are forwarded. This argument is required for multi-access networks. It is not required for serial interface routes.

Examples:

```
clns route 38.0002 serial 0
clns route 38.0003 ethernet 1 0000.0c00.1550
clns route 39.0003 serial 1 4085551212
```

[no] clns route *nsap-prefix {next-hop-NET/name}*

Establishes a specific static route.

nsap-prefix—Prefix of NSAPs to be forwarded.

next-hop-NET—NET where forwarded NSAPs are sent.

name—Name of interface where forwarded NSAPs are sent.

Examples:

```
clns route 47.0004.000c 47.0005.0001.0000.0001.0000.00
clns route 39.0001 serial 0
```

[no] clns route nsap-prefix discard

Tells a router to discard packets with the specified *nsap-prefix*.

nsap-prefix—Prefix of NSAPs to be forwarded.

Example:

```
clns route 47.0004.000c discard
```

[no] router [isis|iso-igrp] area-tag

Identifies the area the router will work in and lets the router know that it will be routing dynamically rather than statically.

isis and **iso-igrp**—The keyword **isis** specifies the IS-IS protocol. The keyword **iso-igrp** specifies the ISO-IGRP protocol.

area-tag—Meaningful name for an area.

Example:

```
router isis area1
```

ISO CLNS Routing Protocol Router Subcommands

[no] distance *value*

Assigns or deletes an administrative distance for a particular routing protocol. The **no** form of the command restores the defaults: static routes = 10, ISO-IGRP routes = 100, IS-IS routes = 110.

[no] is-type [**level-1** | **level-1-2** | **level-2-only**]

Configures the level at which the router should operate. The **no is-type** command resets routing level to the default: Level 1 and 2.

level-1 | **level-1-2** | **level-2-only**—If **level-1** is specified, the router acts as a station router. If **level-1-2** is specified, the router acts as both a station router and a router that routes between areas. If **level-2-only** is specified, the router acts as an inter-area router only.

Example:

```
is-type level-1
```

[no] net *NET*

Configures a Network Entity Title (NET) for the routing process.

NET—Network Entity Title, which defines the area addresses for the ISO-IGRP area.

Example:

```
net 47.0004.004d.0001.0000.0c11.1111.00
```

[no] redistribute [**iso-igrp** | **isis**] *area-tag*

Redistributes routing information throughout a routing area. The **no** form of the command disables this CLNS routing protocol on this interface.

Arguments:

area-tag—Defined tag for the areas from which the routing information is to be redistributed.

Example:

```
redistribute iso-igrp area2
```

[no] redistribute static

Causes the router to inject any static CLNS routes into the domain. The **no** form of the command stops redistribution.

ISO CLNS Routing Protocol Interface Subcommands

[no] clns route *nsap-prefix interface-type unit [SNPA-address | X.121-address]*

Establishes a specific static route for a named interface.

nsap-prefix—Prefix of NSAPs to be forwarded.

interface-type unit—Name and number of interface.

SNPA address or *X.121 address*—Supply the SNPA address, which is required for multi-access networks except for serial interface routes, or the X.121 address.

Examples:

```
clns route 39.0001 serial 0
clns route 38.0002 ethernet 10000.0c00.1550
```

[no] clns router iso-igrp *area-tag [level2]*

Specifies ISO-IGRP routing. The **no** form of the command with an area tag disables ISO IGRP routing for the system.

area-tag—Tag defined for the NET using the **router** global configuration command.

level2—Specifies that the interface can advertise Level 2 information.

Example:

```
clns router iso-igrp area3 level2
```

[no] clns router isis *area-tag*

Enables IS-IS routing for OSI on a specific interface. The **no** form of the command with an area tag disables IS-IS routing for the system.

area-tag—Tag defined for the NET using the **router** global configuration command.

level2—Specifies that the interface can advertise Level 2 information.

Example:

```
clns router isis area1
```

[no] isis circuit-type [**level-1** | **level-1-2** | **level-2-only**]

Configures the type of *adjacency* desired for this interface. The **no** form of the command restores the default: **level-1-2**.

level-1—If **level-1** is specified and there is at least one area address in common between this system and its neighbors, at most a Level 1 adjacency may be established.

level-1-2—If **level-1-2** is specified, a Level 1 and 2 adjacency is established if the neighbor is configured as **level-1-2** and there is at least one area in common. If there is no area in common, a Level 2 is established.

level-2-only—If **level-2-only** is specified, a Level 2 adjacency is established if and only if the neighbor is configured exclusively to be a Level 2 router.

[no] isis metric *default-metric delay-metric expense-metric error_metric*
[level-1 | level-2]

Configures the metric (or cost) for the specified interface.

default-metric—Default metric value. Default: 10.

delay-metric—Not currently supported.

expense-metric—Not currently supported.

error_metric—No currently supported.

level-1 or **level-2**—If you do not specify **level-1** or **level-2**, the default metric applies to Level 1 and Level 2. To reset the metric for Level 1 or Level 2 only, specify the appropriate keyword.

Example:

```
isis metric 8 level-1
```

[no] isis password *password* [level-1 | level-2]

Configures the authentication password for an interface. The **no** form of the command restores the default: disabled.

password—Character string containing up to 80 characters.

level-1 or **level-2**—Indicates the level for which you want to specify the password.

Example:

```
isis password yourpassword level-1
```

[no] isis priority *value* [level-1 | level-2]

Sets the priority for designated router election. Priorities can be configured for Level 1 and Level 2 individually. The **no** form of the command restores the default: 64.

value—Priority value, from 1 to 64.

level-1 or **level-2**—If you do not specify **level-1** or **level-2**, the priority applies to Level 1 and Level 2. To reset the priority for Level 1 or Level 2 only, specify the appropriate keyword.

Example:

```
isis priority 32
```

Novell Global Commands

[no] novell maximum-paths *paths*

Follows the **novell routing** command to set the maximum number of multiple paths that the router will remember and use. The **no** form of the command restores the default: 1.

paths—Number of paths to be remembered.

Example:

```
novell maximum-paths 3
```

[no] novell route *network network.address*

Specifies or removes static routes for a Novell network.

network—The network whose messages you want to forward.

network.address—The network address of the router to which you want to forward network traffic.

Example:

```
novell route 5e 3abc.0000.0c00.1ac9
```

[no] novell routing [*host-address*]

Enables or disables Novell routing and Novell RIP routing and SAP services.

host-address—System-wide host address. If you do not specify an address, the MAC address of the first Ethernet, Token Ring, or FDDI interface is used. The address must not be multicast.

Example:

```
novell routing 0000.0c00.23fe
```

Novell Interface Subcommands

novell encapsulation {**novell-ether** /**arpa**}

Selects which data format or encapsulation is used on an Ethernet interface. Default: Novell IPX over Ethernet using Novell's variant of IEEE 802.2 encapsulation.

Choose one of these keywords:

novell-ether—Enter to specify Novell IPX over Ethernet using Novell's variant of IEEE 802.2 encapsulation.

arpa—Enter when the Novell systems must communicate with other vendors' systems, such as DEC VAX/VMS. In this case, Ethernet-style encapsulation is used with a protocol type of 8137.

Example:

```
novell encapsulation arpa
```

[no] novell helper-address *net.host*

Forwards broadcast packets that match the access list specified by the **novell helper-list** subcommand. Useful for hosts which use a protocol other than SAP to advertise their availability.

net.host—The network and host address to which broadcast packets are forwarded.

Example:

```
novell helper-address 3abc.0000.0c00.1ac9
```

[no] novell helper-list *access-list-number*

Forwards packets that pass the specified Novell access list to the Novell helper host. The **no** form of the command disables the function.

access-list-number—The access list, containing numbers expressed as decimal values.

Example:

```
novell helper-list 801
```

[no] novell network *number*

Enables and disables Novell routing on a particular interface.

number—Number of the Novell network to which the interface is attached. Novell packets received on an interface that do not have a Novell network number are ignored.

Example:

```
novell network 5e
```

[no] novell output-sap-delay *delay*

Establishes a delay between SAP updates so that the Cisco router interface operates at the slower speed of the Novell server. The delay is added to the usual SAP reporting interval. The **no** form of the command disables the mechanism.

delay—The delay between SAP updates, in milliseconds.

Example:

```
novell output-sap-delay 200
```

[no] novell route-cache

Enables or disables Novell fast switching. Default: enabled.

novell sap-interval *interval*

Establishes less frequent SAP updates over slow links. Default: 1 minute.

interval—Number of minutes between SAP updates. If *interval* is 0, periodic updates are not sent—a message is sent only when the server first appears and when it goes down.

Example:

```
novell sap-interval 5
```

[no] novell source-network-update

When enabled, repairs corrupted network numbers by setting the source network field of any packet with a hop count of zero to the local network number.

[no] novell update-time *seconds*

Allows the Novell routing update timers to be set individually for each interface. This command can be used only in an all-Cisco environment, and all timers should be the same for routers connected to the same network segment. The **no** form of the command restores the default: 30 seconds.

seconds—The interval between updates. Minimum: 10.

Example:

```
novell update-time 40
```

Novell Access List and Filtering Commands

[no] access-list *number* {**deny** | **permit**} *novell-source-network* [*source-address* [*source-mask*]] *novell-destination-network* [*destination-address* [*destination-mask*]]

Global configuration command that specifies standard Novell IPX access lists. The **no** form of the command removes any access list in the current image with the specified number.

number—A unique number for this access list, from 800 to 899.

deny or **permit**—Specifies the deny or permit condition.

novell-source-network—Source network.

source-address—Address of the source network.

source-mask—Mask for the *source-address*.

novell-destination-network—Destination network.

destination-address—Address of the destination network.

destination-mask—Mask for the *destination-address*.

Examples:

```
access list 800 deny -1 2
access list 800 deny 1.0000.0c00.1111
access-list6 800 permit 1.1111.1111.1111 0000.0000.0000
2.2222.2222.2222 0000.0000.0000
```

[no] access-list *number* {**deny**|**permit**} *novell-protocol* *source-network* [*source-address* [*source-mask*]] *source-socket* *destination-network* [*destination-address* [*destination-mask*]]*destination-socket*

Global configuration command that specifies extended Novell IPX access lists. The **no** form of the command removes any access list in the current image with the specified number.

number—A unique number for this extended access list from 900 to 999.

deny or **permit**—Specifies the deny or permit condition.

novell-protocol—Novell protocol number.

source-network—Source network.

source-address—Address of the source network. A network number of -1 matches all networks.

source-mask—Mask for the *source-address*.

source-socket—Socket number. A socket number of zero matches all sockets.

destination-network—Destination network.

destination-address—Address of the destination network.

destination-mask—Mask for the *destination-address*.

destination-socket—Destination socket number. A socket number of zero matches all sockets.

Examples:

```
access list 900 deny 1 1 1234 2 1234
access list 900 deny 1 1.1111.1111.1111 0000.0000.0000
1234 2.2222.2222.2222 0000.0000.0000 1234
```

[no] access-list *number* {**permit** | **deny**} *network*[.*address*] [*service-type*]

Global configuration command that defines an access list for filtering SAP requests.

number—A unique decimal number from 1000 to 1099.

permit or **deny**—Specifies the type of access desired. Permit or deny access is based on the data provided.

network—A hexadecimal Novell network number. The value zero defines the local network; the value -1 defines all networks.

address—Novell host address.

service-type—The service type to filter, in hexadecimal format. The value zero represents all services.

Example:

```
access-list 1001 deny -1 4
access-list 1001 permit -1
```

[no] novell access-group *number*

Interface subcommand that assigns a Novell IPX access list group number to a specific interface. The **no** form of the command removes the number.

number—Novell access list number. All outgoing packets forwarded through the interface will be filtered by this access list.

Example:

```
novell access-group 815
```

[no] novell input-network-filter *access-list-number*

Interface subcommand that explicitly specifies the networks that are added to the Novell IPX routing table. The **no** form of the command disables the function.

access-list-number—The access list number specified in the **novell access-list** command.

Example:

```
novell input-network-filter 801
```

[no] novell input-sap-filter *access-list-number*
[no] novell output-sap-filter *access-list-number*

Global configuration commands that configure Cisco routers to filter Novell SAP messages. The **no** forms of the commands remove the filters.

access-list-number—A SAP Novell access list from 1000 to 1099. Used with the **novell input-sap-filter** command, the number defines the acceptable source of Novell SAP messages. Used with the **novell output-sap-filter** command, the number defines the intended destination of SAP messages.

Example:

```
novell input-sap-filter 1000
```

[no] novell output-network-filter *access-list-number*

Interface subcommand that explicitly specifies the list of networks that are sent in routing updates. The **no** form of the command disables the function.

access-list-number—The access list number specified in the **novell access-list** command.

Example:

```
novell output-network-filter 821
```

[no] novell router-filter *access-list-number*

Interface subcommand that specifies or removes the list of routers from which data will be accepted.

access-list-number—Access list number specified in the **novell access-list** command.

Example:

```
novell router-filter 823
```

[no] novell router-sap-filter *access-list-number*

Global configuration command that configure Cisco routers to filter Novell SAP messages. The **no** form of the command removes the filters.

access-list-number—A SAP Novell access list from 1000 to 1099. Defines the specific router from which SAP filters will be accepted.

Example:

```
novell router-sap-filter 1000
```

PUP Commands

[no] pup routing

Enables or disables PUP routing.

[no] pup *address*

Enables or disables PUP routing on a particular interface.

address—The desired PUP address of the interface.

Example:

```
pup 10#371
```

pup map *address*

Maps a PUP network onto one subnetted IP network.

address—Network number of the major network over which the PUP network is overlaid.

Example:

```
pup map 10#371
```

[no] pup helper-address *address*

Sets the helper address so that broadcasts to the PUP Miscellaneous Services socket can be forwarded to another network (the helper address) where the appropriate servers can be found.

address—The PUP address of the desired server.

Example:

```
pup helper-address 10#371
```

VINES Global Configuration Commands

[no] vines host *name address*

Adds or deletes an entry to the VINES name-to-address mapping table. Once entered, a VINES name can be used anywhere that a VINES address is requested.

name—The name of a VINES host. It can be any length.

address—The VINES address, expressed in the form *network:subnet*.

Example:

```
vines host STUFF 0027AF92:8001
```

[no] vines routing [*address*]

Enables or disables VINES routing.

address—Specifies the VINES address of the router. The format is *network:subnet*. The *subnet* must always be 1. The *address* argument is not necessary for Ethernet interfaces because the router automatically maps itself into the VINES address space that is reserved for Cisco.

Example:

```
vines routing 3001082D:1
```

VINES Interface Subcommands

[no] vines arp-enable

Enables or disables the processing of ARP packets received on the defined interface. When enabled, the router responds to ARP packets and assigns network addresses to clients.

vines encapsulation [arpa | snap | vines-tr]

Sets the MAC-level encapsulation used for VINES broadcasts on the defined interface. The current defaults are: **arpa** for Ethernet lines, **snap** for IEEE 802.2 media, and **vines-tr** for Token Rings.

[no] vines metric [number]

Enables or disables the processing of VINES processing on this interface.

number—The VINES delay metric, which is used in routing updates. If this argument is not supplied, the system automatically chooses a reasonable metric value based on the delay value set for the defined interface.

[no] vines redirect-interval [number]

Determines how frequently the router sends an ICP *redirect* message on a given interface.

number—The minimum number of seconds that must pass before the router sends *redirect* messages. If the value for *number* is 0, the router will never send redirects on this interface.

Example:

```
vines redirect-interval 0
```

[no] vines serverless

Configures a Banyan VINES network without a server. This command can be used on several routes to build a path to a network that contains servers. The **no** form of the command restores the default: **no vines**

serverless.

VINES Access List Commands

[no] vines access-group *list*

Interface subcommand that assigns or removes an outgoing VINES access list to the defined interface. The **no** form of the command removes the access list.

list—Number of an access list defined with the **vines access-list** command.

Example:

```
vines access-group 2
```

[no] vines access-list *number* {**permit**|**deny**} **IP** *source-address source-mask dest-address dest-mask*

[no] vines access-list *number* {**permit**|**deny**} *protocol source-address source-mask source-port dest-address dest-mask dest-port*

Global configuration command that creates or deletes a VINES access list.

number—Access list number from 1 to 100.

permit or **deny**—Specifies the permit or deny condition.

protocol—An integer between 1 and 255, or one of the following protocols:

- **SPP**—Sequence Packets Protocol
- **RTP**—Routing Table Protocol
- **ARP**—Address Resolution Protocol
- **IPC**—Interprocess Communications
- **ICP**—Internet Control Protocol

source-address and *dest-address*—VINES addresses, written in the standard form.

source-mask and *dest-mask*— VINES addresses that indicate the bits in the corresponding address that should be ignored.

source-port and *dest-port*—Port numbers, expressed as integers from 0 to 65535.

Examples:

```
vines access-list 1 permit IP 0:1 ffffffff:0 0:1 ffffffff:0
vines access-list 1 deny IP 0:0ffffffff:ffff0:0ffffffff:ffff
no vines access-list 1
```

XNS Global Configuration Commands

[no] xns forward-protocol *type*

Defines the protocol types that will be forwarded when a broadcast is received on an interface that has an XNS helper address. The **no** form of the command disables forwarding of the specified protocol.

type—A decimal number corresponding to an appropriate XNS protocol.

Example:

```
xns forward-protocol 2
```

[no] xns maximum-paths *paths*

Sets the maximum number of equal-cost paths the router will use.
Default: 1.

paths—Maximum number of paths.

Example:

```
xns maximum-paths 2
```

[no] xns route *network host-address*

Adds a static route to the XNS routing table.

network—The destination XNS network number, expressed in decimal format.

host-address—A decimal XNS network number and the hexadecimal host number.

Example:

```
xns route 25 51.0456.acd3.1243
```

[no] xns routing [*address*]

Enables XNS routing. The **no** form of the command disables all XNS packet processing.

address—XNS address the router should use. If the argument *address* is omitted, the router uses the first Token Ring, FDDI, or Ethernet interface hardware address found.

Example:

```
xns routing 0123.4567.abcd
```

[no] xns ub-emulation

Enables Ungermann-Bass Net/One routing. Causes Hello packets and routing updates in Ungermann-Bass format to be sent out through all the interfaces on which XNS is enabled. Also causes the Cisco router to get its routing information for remote networks from Ungermann-Bass updates. The **no** form of the command restores the router to standard XNS mode.

[no] xns ub-routing

Makes the router behave exactly as it would if you issued the **xns ub-emulation** command for the system as a whole, as well as the **xns hear-rip**, **xns flood broadcast allnets**, and **xns flood specific allnets** commands for all interfaces on which XNS is enabled.

XNS Interface Subcommands

xns encapsulation {**snap** | **ub** | **3com**}

Selects encapsulation for a Token Ring interface. Default: **snap**.

snap—Token Ring in an IBM installation.

ub—Ungermann-Bass.

3com—Older 3Com Corporation products.

Example:

```
xns encapsulation ub
```

[no] xns flood broadcast allnets

Floods packets with destinations of -1.FFFF.FFFF.FFFF.

[no] xns flood broadcast net-zero

Floods packets with destinations of 0.FFFF.FFFF.FFFF.

[no] xns flood specific allnets

Floods packets with destinations of -1.

[no] xns hear-rip [*access-list*]

Configures the Cisco router to receive RIP updates on some interfaces as if they were Ungermann-Bass updates.

access-list—XNS access list. Include this argument when you only want certain routes to be learned through standard RIP.

[no] xns helper-address *host-address*

Sets a helper address to forward broadcasts. The **no** form of the command disables XNS helpering.

host-address—Combination of the network and host addresses, in dotted decimal format.

Example:

```
xns helper-address -1.FFFF.FFFF.FFFF
```

[no] xns network *number*

Assigns a decimal XNS network number to an interface, enabling that interface to run XNS protocols.

number—Network number, in decimal format.

Example:

```
xns network 20
```

[no] xns route-cache

Enables or disables fast switching.

xns update-time *seconds*

Sets the XNS routing update timers. Use this command only in an all-Cisco environment. Make sure that all timers are the same for routers attached to the same network segment. Default: 30

seconds—How often RIP sends routing table updates to its neighbor routers. Must be greater than or equal to 10 seconds.

Example:

```
xns update-time 20
```

XNS Access List and Network Filtering Commands

[no] access-list *number* { **permit** | **deny** } *XNS-source-network* [. *source-address* [*source-mask*]] *XNS-destination-network* [. *destination-address* [*destination-mask*]]

Global configuration command that configures or removes an XNS access list.

number—Access list number in the range 400 to 499.

permit or **deny**—Specifies the filtering action.

XNS-source-network—The XNS source network.

source-address—Address of the *XNS-source-network*.

source-mask—Mask for the *source-address*.

XNS-destination-network—The XNS destination network.

source-address—Address of the *XNS-destination-network*.

source-mask—Mask for the *destination-address*.

Examples:

```
access-list 400 deny -1 2
access-list 400 deny 21.0000.0c00.1111
access-list 400 deny 21.011.1622.0015 0000.0000.0000
31.01D3.020C.0022 0000.0000.0000
```

[no] access-list *number* {**deny**|**permit**} *XNS-protocol* *source-network* [*source-address* [*source-mask*]] *source-socket* *destination-network* [*destination-address* [*destination-mask*]] *destination-socket*

Global configuration command that configures or removes an extended XNS access list.

number—Access list number in the range 500 to 599.

permit or **deny**—Specifies the filtering action.

XNS-protocol—XNS protocol number.

XNS-source-network—The XNS source network.

source-address—Address of the *XNS-source-network*.

source-mask—Mask for the *source-address*.

XNS-destination-network—The XNS destination network.

source-address—Address of the *XNS-destination-network*.

source-mask—Mask for the *destination-address*.

Examples:

```
access-list 400 deny 1 1 1234 2 1234
access-list 400 deny 1 21.110011.1622.001500.0000.0000
31.01D3.020C.0022 0000.0000.0000. 1234
```

no access-list *number*

Removes the specified access list.

number—Access list number.

Example:

```
no access-list 401
```

[no] xns access-group *number*

Interface subcommand that assigns an access list number to an interface. Also compares all XNS packets that ordinarily would have been forwarded by the router through this interface to the access list. If the packet fails the access list, it is discarded and not transmitted.

number—Access list number.

Example:

```
xns access-group 500
```

[no] xns input-network-filter *access-list-number*

Interface subcommand that defines the networks that are added to your router's routing table.

access-list-number—Access list number specified in the XNS **access-list** command.

Example:

```
xns input-network-filter 4
```

[no] xns output-network-filter *access-list-number*

Interface subcommand that defines the list of networks that are sent out in routing updates by your router.

access-list-number—Access list number.

Example:

```
xns output-network-filter 4
```

[no] xns router-filter *access-list-number*

Interface subcommand that controls the list of routers from which data will be accepted.

access-list-number—Access list number specified in the XNS **access-list** command.

Example:

```
xns router-filter 466
```


Bridging Configuration

Transparent Bridging Global Configuration Commands

[no] bridge *group* acquire

Global configuration command that enables or disables the dynamic learning process. Default: disabled.

group—Spanning tree group number.

Example:

```
bridge 1 acquire
```

[no] bridge *group* domain *domain-number*

Enables or disables multiple domain spanning trees. Only those devices in the domain can share spanning tree information. This command works only when the bridge group is running the IEEE spanning tree protocol. Non-Cisco bridges might not work correctly on networks containing Cisco bridges domain numbers other than zero.

group—Bridge group number between 0 and 10, as specified in the **bridge *group* protocol ieee** command.

domain-number—Unique domain number you choose. The domain number zero is required for communicating with IEEE bridges that do not support this domain extension. Default: zero.

Example:

```
bridge 1 domain 3
```

bridge *group* forward-time *seconds*

Sets the default the forward delay interval, that is, the amount of time the bridge listens for topology change information after an interface has been activated for bridging and before forwarding actually begins. Default: 30 seconds.

group—Spanning tree group number.

seconds—The forward delay interval, from 10 to 200 seconds.

Example:

```
bridge 1 forward-time 60
```

bridge *group* hello-time *seconds*

Specifies the interval between Bridge Protocol Data Units (BPDUs).
Default: 1 second.

group—Spanning tree group number.

second—The interval, from 1 to 10 seconds.

Example:

```
bridge 2 hello-time 5
```

bridge *group* max-age *seconds*

Specifies the interval that the bridge waits to hear BPDUs from the root bridge before recomputing the bridge spanning tree topology. Default: 15 seconds.

group—Spanning tree group number.

second—Interval the bridge will wait to hear BPDUs from the root bridge.

Example:

```
bridge 2 max-age 20
```

[no] bridge *group* multicast-source

Allows or disallows the forwarding of frames with multicast source addresses. This command does not affect the learning of frames.

group—Spanning tree group number.

Example:

```
bridge 2 multicast-source
```

bridge group priority number

Sets the priority of an individual bridge for selection as the root bridge. A lower number increases the likelihood for selection as the root bridge. Minimum = 1, maximum = 65000, default = 128.

group—Spanning tree group number.

number—Priority number from 1 to 65000.

Example:

```
bridge 2 priority 1000
```

[no] bridge group protocol {dec | ieee}

Defines or removes a spanning tree protocol and spanning tree group.

group—Spanning tree group number, from 1 to 9.

ieee or **dec**—The protocol to use.

Example:

```
bridge 9 protocol dec
```

Transparent Bridging Interface Subcommands

[no] bridge-group group

Assigns the network interface to the spanning tree group. The **no** form of the command removes the network interface.

group—Spanning tree group number.

Example:

```
bridge-group 2
```

bridge-group group circuit number

Establishes load balancing, marking a serial interface as belonging to circuit group number. Parallel serial interfaces on both bridges must all be flagged as being members of the same circuit group.

group—Spanning tree group number.

number—Circuit group number, expressed as an integer from 1 to 9.

Example:

```
bridge-group 3 circuit 1
```

[no] bridge-group *group* lat-compression

Reduces the amount of bandwidth LAT traffic consumes on serial interfaces. LAT compression may be specified for serial compression only.

group—Spanning tree group number.

Example:

```
bridge-group 2 lat-compression
```

[no] bridge-group *group* path-cost *cost*

Sets or removes a different path cost. Default: 100

group—Spanning tree group number.

cost—Integer from 0 to 65535. Higher values indicate higher costs.

Example:

```
bridge-group 2 path-cost 20
```

bridge-group *group* priority *number*

Assigns a priority to an interface. This priority is used in tie-breaking when computing a network topology. Default: 0.

group—Spanning tree group number.

number—Priority value, expressed as an integer from 0 to 255. The lower the number, the more likely it is that the bridge on this interface will be chosen as the root.

Example:

```
bridge-group 1 priority 10
```

Transparent Bridging Access List and Filtering Commands

[no] access-list *list* {**permit** | **deny**} *address-mask*

Global configuration command that prepares access control information for filtering of frames by canonical (Ethernet ordered) MAC address. The **no** form of the command removes a single access list entry.

list—An integer from 700 to 799 selected for the list.

address—48-bit canonical MAC address written in dotted triplet form.

mask—48-bit canonical MAC address written in dotted triplet form.

The ones bits in the *mask* argument are the *address* bits to be ignored.

Examples:

```
access-list 700 deny 0800.2000.0000 0000.00FF.FFFF
access-list 700 permit 0000.0000.0000 FFFF.FFFF.FFFF
```

[no] access-list *list* {**permit** | **deny**} *type-code wild-mask*

Global configuration command that prepares access control information for filtering of frames by protocol type. The **no** form of the command removes a single access list entry.

list—Unique number from 200 to 299 that identifies the list.

permit or **deny**—Indicates whether the frame is permitted or denied.

type-code—16-bit hexadecimal number with a leading “0x.”

wild-mask—16-bit hexadecimal number whose ones bits correspond to bits in the *type-code* argument that the router should ignore when making a comparison.

Examples:

```
access-list 201 permit 0x6005 0x0000
access-list 201 deny 0x0000 0xFFFF
```

[no] bridge *group address* *MAC-address* [**forward** | **discard**] [*interface*]

Global configuration command that adds or removes an address from the forwarding database. The **no** form of the command followed by the MAC address removes an address from the forwarding database.

group—Spanning tree group number.

MAC-address—48-bit dotted triplet canonical (Ethernet ordered) hardware address such as those displayed by the EXEC **show arp** command. The address can be a station address, the broadcast address, or a multicast destination address.

forward—Enables the bridge to forward a frame sent from or destined for the specified address.

discard—Instructs the bridge to discard frames sent from or destined for the specified address without further processing.

interface—Specifies an interface on which the address can be reached.

Example:

```
bridge 1 address 0800.cb00.45e9 forward ethernet 1
```

[no] **bridge group lat-service-filtering**

Global configuration command that enables or disables LAT service filtering. The **no** form of the command restores the default: disabled.

group—Bridge group.

[no] **bridge-group group input-address-list list**

Interface subcommand that assigns or removes an access list to or from a particular interface for filtering by the MAC source addresses.

group—Spanning tree group number.

list—Access list number between 200 and 299, assigned with the bridge **access-list** command.

Example:

```
bridge-group 1 input-address-list 200
```

[no] bridge-group *number* **input-lat-service-deny** *group**list*

Interface subcommand that specifies the group codes by which to deny access upon input. Default: no filtering.

number—Spanning tree group number.

*group**list*—Specifies the LAT groups. Single numbers and ranges are permitted.

Example:

```
bridge-group 1 input-lat-service-deny 1 5 12-14
```

[no] bridge-group *number* **input-lat-service-permit** *group**list*

Interface subcommand that specifies the group codes by which to permit access upon input. Default: no filtering.

number—Spanning tree group number.

*group**list*—Specifies the LAT groups. Single numbers and ranges are permitted.

Example:

```
bridge-group 1 input-lat-service-permit 1 5 12-14
```

bridge-group *group* **input-lsap-list** *list*

Interface subcommand that adds or removes a filter for IEEE 802-encapsulated packets on input. This access list is applied to all IEEE 802 frames received on that interface prior to the bridge-learning process.

group—Spanning tree group number.

list—Access list number between 200 and 299, assigned with the bridge **access-list** command.

Example:

```
bridge-group 2 input-lsap-list 250
```

bridge-group *group* **input-type-list** *list*

Interface subcommand that adds or removes a filter for Ethernet- and SNAP-encapsulated packets on input. The bridge applies the access list to all Ethernet frames received on that interface prior to the bridge learning process.

group—Spanning tree group number.

list—Access list number between 200 and 299, assigned with the bridge **access-list** command.

Example:

```
bridge-group 4 input-type-list 200
```

[no] bridge-group *group* **output-address-list** *list*

Interface subcommand that assigns or removes an access list to or from a particular interface for filtering by the MAC destination addresses.

group—Spanning tree group number.

list—Access list number between 200 and 299, assigned with the bridge **access-list** command.

Example:

```
bridge-group output-address-list 204
```

bridge-group *number* **output-lat-service-deny** *group**list*

Interface subcommand that specifies the group codes by which to deny access upon output. Default: no filtering.

number—Spanning tree group number.

*group**list*—Specifies the LAT groups. Single numbers and ranges are permitted.

Example:

```
bridge-group 1 output-lat-service-deny 1 5 12-14
```

bridge-group *number* output-lat-service-permit *grouplist***

Interface subcommand that specifies the group codes by which to permit access upon output. Default: no filtering.

number—Spanning tree group number.

*group**list*—Specifies the LAT groups. Single numbers and ranges are permitted.

Example:

```
bridge-group 1 output-lat-service-permit 1 5 12-14
```

bridge-group *group* output-lsap-list *list*

Interface subcommand that adds or removes a filter for IEEE 802-encapsulated packets on output. This access list is applied just before sending out a frame to an interface.

group—Spanning tree group number.

list—Access list number between 200 and 200, assigned with the bridge **access-list** command.

Example:

```
bridge-group 3 output-lsap-list 223
```

bridge-group *group* output-type-list *list*

Interface subcommand that adds or removes a filter for Ethernet- and SNAP-encapsulated packets on output. The bridge applies the access list just before sending out a frame to an interface.

group—Spanning tree group number.

list—Access list number between 200 and 299, assigned with the bridge **access-list** command.

Example:

```
bridge-group 5 output-type-list 299
```

IBM Connectivity

LLC2 Interface Subcommands

llc2 ack-delay-time *msec*

Controls the maximum amount of time the Cisco router allows incoming I-frames to stay unacknowledged. Minimum = 1, maximum = 60000, default = 3200.

msec—Number of milliseconds the Cisco router allows incoming I-frames to stay unacknowledged, even if the router has not received the number of frames specified with the **llc2 ack-max** command.

Example:

```
llc2 ack-delay-time 800
```

llc2 ack-max *packet-count*

Controls the maximum number of information frames (I-frames) received by the Cisco router before it must send an acknowledgment to these frames. Minimum = 1, maximum = 255, default = 3.

packet-count—The number of frames.

Example:

```
llc2 ack-max 5
```

llc2 idle-time *msec*

Controls the frequency of polls during periods of idle traffic. Minimum = 1, maximum = 60000, default = 10000.

msec—The number of milliseconds that pass when there is no traffic before the LLC2 station sends a Receiver Ready frame.

Example:

```
llc2 idle-time 1500
```

llc2 local-window *packet-count*

Controls the maximum number of information frames sent by the router before it waits for an acknowledgment to these frames. Minimum = 1, maximum = 127, default = 7.

packet-count—Maximum number of I-frames sent before router must wait for an acknowledgment.

Example:

```
llc2 local-window 5
```

llc2 n2 *retry-count*

Controls the number of times the Cisco router retries operations such as sending an unacknowledged frame or polling a remote busy station. Minimum = 1, maximum = 255, default = 8.

retry-count—Number of times router should retry various operations.

Example:

```
llc2 n2 5
```

llc2 t1-time *msec*

Controls how long the Cisco router waits for an acknowledgment to transmitted I-frames. Minimum = 1, maximum = 60000, default = 1000.

msec—Maximum number of milliseconds the T1-timer should wait for an acknowledgment from the receiver that an I-frame has been received.

Example:

```
llc2 t1-time 10000
```

llc2 tbusy-time *msec*

Controls the amount of time the Cisco router waits while the other LLC2 station is in a busy state before attempting to poll the remote station again. Minimum = 1, maximum = 60000, default = 9600.

msec—Number of milliseconds that the Cisco router waits.

Example:

```
llc2 tbusy-time 10000
```

llc2 tpf-time *msec*

Controls the amount of time the router waits for a final response to a poll frame that it sent before the router re-sends the original poll frame. Minimum = 1, maximum = 60000, default = 1000.

msec—Number of milliseconds the router waits.

Example:

```
llc2 tpf-time 10000
```

llc2 trej-time *msec*

Controls the amount of time the Cisco router waits for a resend of a rejected frame before sending the reject (REJ) command to the remote station. Minimum = 1, maximum = 60000, default = 3200.

msec—Number of milliseconds that the Cisco router waits.

Example:

```
llc2 trej-time 30000
```

llc2 xid-neg-val-time *msec*

Controls the frequency of exchange of identification (XID) frame transmissions by the Cisco router. Cisco Systems recommends that you do not change this parameter unless asked to by Cisco technical personnel. Minimum = 1, maximum = 60000, default = 0.

msec—Number of milliseconds after which the Cisco router transfers XID frames to other LLC2-speaking stations.

Example:

```
llc2 xid-neg-val-time 10
```

llc2 xid-retry-time *msec*

Controls how long the router waits for a reply to the exchange of identification (XID) frames it sends to remote stations.

Minimum = 1, maximum = 60000, default = 60000.

msec—Maximum number of milliseconds the Cisco router should wait. The *msec* value for this command should be larger than the *msec* value in the **t1-time** command.

Example:

```
llc2 xid-retry-time 10000
```

SDLC Interface Subcommands

encapsulation sdlc-primary

Identifies an interface that requires an SDLC configuration.

[no] sdlc address *hexbyte*

Assigns the set of secondary stations attached to the serial link.

hexbyte—The hexadecimal address of the secondary station.

Example:

```
sdlc address c1
```

[no] sdlc fair-poll-timer *msec*

Specifies how long the Cisco router can continue to send output before it begins polling the secondary station for input. The **no** form of the command restores the default: 500 milliseconds. Minimum = 100, maximum = 10000.

msec—Number of milliseconds the Cisco router has for uninterrupted sending of output.

Example:

```
sdlc fair-poll 750
```

[no] sdlc frmr-disable

Indicates whether the secondary devices off this link support Frame Rejects (FRMRs). The **no** version of the command restores the default: enabled.

sdlc k *window-size*

Controls the local send window size of the Cisco router. Minimum = 1, maximum = 7, default = 7.

window-size—The maximum number of I-frames the Cisco router receives before the other end must stop sending and wait for an acknowledgment.

Example:

```
sdlc k 7
```

sdlc holdq *address queue-size*

Controls the maximum number of outstanding packets that may be held for transmission to a remote SDLC device. Minimum = 1, maximum = none, default = 12.

address—The SDLC address for which you want to specify a queue size.

queue-size—The maximum number of outstanding packets held for transmission.

Example:

```
sdlc holdq c1 30
```

sdlc n1 *bit-count*

Controls the maximum number of bits allowed for incoming frames on this link. Minimum = 1, maximum = 12000, default = 12000.

bit-count—Maximum bit size. Frames that exceed this size are rejected.

Example:

```
sdlc n1 3000
```

sdlc n2 *retry-count*

Controls the number of times the Cisco router attempts to retry an operation that has timed out. Minimum = 1, maximum = 255, default = 20.

retry-count—Number of retry attempts. When this number is exceeded, the SDLC station terminates its session with the other station.

Example:

```
sdlc n2 5
```

[no] sdlc poll-limit-value *count*

Controls how many times the Cisco router can poll a particular secondary station before it must poll the next station in the poll list. The **no** form of the command restores the default: 1. Minimum = 1, maximum = 10.

count—Maximum number of times the Cisco router can poll one station before proceeding to the next.

Example:

```
sdlc poll-limit-value 3
```

[no] sdlc poll-pause-timer *msec*

Controls how long the router pauses between sending each series of poll frames. The **no** version of the command restores the default: 100 milliseconds. Minimum = 100, maximum = 10000.

msec—Number of milliseconds that the router waits before sending.

Example:

```
sdlc poll-pause timer 150
```

sdlc t1 *timeout*

Controls the amount of time the Cisco router waits for a reply to a frame or sequence of frames. Minimum = 1, maximum = 64000, default = 3000.

timeout—The time, in milliseconds, that the Cisco router waits.

Example:

```
sdlc t1 6000
```

Source-Route Bridge Global Configuration Commands

[no] lnm snmp-only

Prevents any LNM stations from modifying parameters in the Cisco router. The **no** form of the command allows modifications.

rif *MAC-address* [*RIF-string*] [*interface-name*] **ring-group** *ring*]

no rif *MAC-address* [*interface-name*] **ring-group** *ring*]

Inserts or removes an entry into the RIF cache.

MAC-address—12-digit hexadecimal string written as a dotted triplet.

RIF-string—Series of four-digit hexadecimal numbers, each series separated by a dot (.).

interface-name—The interface from which this RIF entry would have arrived.

ring-group *ring*—The ring group number from which this RIF entry would have arrived.

Example:

```
rif 1000.5A01.0203 0830.0155.100a.5550
```

[no] rif timeout *minutes*

Defines the period of inactivity allowed before unused RIF cache entries are removed. The **no** form of the command resets the RIF timeout period to the default: 15 minutes.

minutes—Number of minutes.

Example:

```
rif timeout 20
```

[no] source-bridge enable-80d5

Enables or disables the conversion of frames between Token Ring LLC2 and Ethernet Type II 0x80d5 format. When disabled, Token Ring LLC2 frames are converted to IEEE 802.3 format LLC2 frames. This command

applies only to frames going through the Source-Route Translational Bridge (SR/TLB), and not to frames going across a Source-Route Transparent (SRT) bridge.

[no] source-bridge largest-frame *ring-group size*

Defines the largest frame size to communicate with all peers in the ring group.

ring-group—Ring group number.

size—Maximum frame size. The legal values for this argument are 516, 1500, 2052, 4472, 8144, 11407, and 17800 bytes.

Example:

```
source-bridge 5 2052
```

[no] source-bridge old-oui

Causes the OUI code in Token Ring frames translated to and from Ethernet Type II to be 0x000000. If you want to transfer data between IBM 8209 Ethernet/Token Ring bridges and Cisco routers running the SR/TLB software, you must issue this global command on each Cisco router. The **no** form of the command restores the default value: 0x0000F8.

[no] source-bridge passthrough *ring-number*

Specifies that frames destined to ring number should never be terminated with Local Acknowledgment. The **no** form of the command removes this restriction.

ring-number—Ring number.

Example:

```
source-bridge passthrough 5
```

[no] source-bridge remote-peer *ring-group interface interface-name*
[if size] **[version number]**

Defines or removes a serial interface over which to run bridged Token Ring traffic.

ring-group—Ring group number.

interface-name—Name of the serial interface.

If size—Largest frame size to send to the remote peer.

version number—Specifies the forced RSRB protocol version number for the remote peer.

Example:

```
source-bridge remote-peer 5 interface serial0
```

[no] source-bridge remote-peer *ring-group tcp ip-address* [**If size**]
[local-ack] [version number]

Defines or removes a remote peer for the specified ring group.

ring-group—Ring group number.

ip-address—IP address.

If size—Largest frame size to send to the remote peer.

local-ack—Specifies that Local Acknowledgment should be used for LLC2 session going to this remote peer.

version number—Specifies the forced RSRB protocol version number for the remote peer.

Example:

```
source-bridge remote-peer 5 tcp 131.108.2.29
```

[no] source-bridge ring-group *ring-number*

Establishes or removes a ring group.

ring-group—Ring group number.

Example:

```
source-bridge ring-group 5
```

[no] source-bridge sap-80d5 *SAP*

When used in conjunction with the **source-bridge enable-80d5** command, enables or disables the translation of Token Ring LLC2 frames to Ethernet Type 2 80d5 format frames.

If the **source-bridge enable-80d5** command is not issued, this command has no effect. You may issue multiple commands, one SAP per line.

SAP—Destination service access point (SAP).

Example:

```
source-bridge sap-80d5 1c
```

[no] source-bridge tcp-queue-max *number*

Sets the maximum output TCP queue length, in packets, that the router will accept for routing to remote source-route bridge peers. The **no** version of the command restores the default of 100.

number—Number of packets.

Example:

```
source-bridge tcp-queue-max 125
```

[no] source-bridge transparent *ring-group pseudo-ring bridge-number tb-group*

Establishes a Source-Route Translational Bridging (SR/TLB) link between the source-bridged *ring-group* and the transparent bridge group *tb-group*.

ring-group—Virtual ring group created by the source-bridge ring-group command.

pseudo-ring—Unique ring number used to represent the transparent bridging domain to the source-route bridging domain.

bridge-number—Pseudo-bridge that attaches the *ring-group* to the *pseudo-ring*.

tb-group—Number of the transparent bridge group that you want to tie into your source-route bridging domain.

Example:

```
source-bridge transparent 10 3 1 1
```

Source-Route Bridge Interface Subcommands

[no] **lnm alternate** *number*

Enables an LRM other than the default LRM to change parameters. The **no** form of the command restores the default: zero.

number—Integer between 0 and 3.

Example:

```
lnm alternate 2
```

[no] **lnm crs**

Enables the Configuration Report Server (CRS), which keeps track of the current logical configuration of a Token Ring. Reports any changes to LNM. Also reports on various other activities such as the change of the Active Monitor on a Token Ring.

[no] **lnm loss-threshold** *number*

Configures the threshold of dropped frames when the router sends a message to all attached LNM. Default: 10 (.1%)

number—Number of frames dropped compared to number of frames forwarded, expressed as hundredths of a percent. Range: 0–9999.

Example:

```
lnm loss-threshold 2
```

[no] **lnm password** *number string*

Assigns a password to a reporting link to prevent unauthorized access from an LRM to a bridge, and to control access to the different reporting links.

number—Identifies the reporting link to which to apply the password.

string—Password containing 6–8 characters: all letters or all numbers. Both letters and numbers can include the characters @, #, \$, or %.

Example:

```
lnm password 4 yourpassword
```

[no] lnm rem

Enables or disables the Ring Error Monitor (REM), which monitors errors reported by any station on the ring. Also monitors whether the ring is in a functional state or in a failure state.

[no] lnm rps

Enables or disables the Ring Parameter Server (RPS), which ensures that all stations on a ring are using a consistent set of reporting parameters. Reports to LNM when any new station joins a Token Ring.

[no] lnm softerr *number*

Controls the frequency of error reports sent from stations on a Token Ring to the Ring Error Monitor. The **no** form of the command restores the timer value to its default: 200 (two seconds).

number—Number of tens of milliseconds between error messages.

Example:

```
lnm softerr 100
```

mac-address *ieee-address*

Sets the MAC layer address. Forces the use of a different MAC address on the specified interface, thereby avoiding the TI MAC firmware problem. It is up to the network administrator to ensure that no other host on the network is using that MAC address.

ieee-address—48-bit IEEE MAC address written as a dotted triple of four-digit hexadecimal numbers.

Example:

```
mac-address 79c1.39de.0003
```

[no] multiring {*protocol-keyword* | **all** | **other**}

Enables or disables the ability of the specified interface to collect and use source-route (RIF) information for routable protocols.

protocol-keyword, **all**, or **other**—To enable the multiring for a single protocol, use one of the following keywords: **apollo**, **appletalk**, **clns**, **decnet**, **ip**, **vines**, or **xns**. To enable the multiring for all frames, use **all**. To enable the multiring for a frame not included in the list, use **other**.

Example:

```
multiring clns
```

[no] source-bridge *local-ring bridge-number target-ring*

Enables or disables source bridging on a specific interface.

local-ring—Ring number from 1 to 4095 for this interface's Token Ring.

bridge-number—Decimal number from 1 to 15 that uniquely identifies a bridge connecting the two rings.

target-ring—Decimal ring number of the destination ring on this router/bridge. It must also be unique within the bridged Token Ring network.

Example:

```
source-bridge 129 1 130
```

[no] source-bridge max-hops *count*

Limits the maximum number of source-route bridge hops of your network. The **no** form of the command restores the count to the maximum value.

count—Defines the number of bridges an explorer packet may traverse.

Example:

```
source-bridge max-hops 7
```

[no] source-bridge old-sna

Enables or disables a workaround for some source-route bridging behavior exhibited by older SNA nodes.

[no] source-bridge proxy-explorer

Enables or disables the proxy explorer function. Default: disabled.

[no] source-bridge route-cache

Enables fast switching to allow for faster implementations of local source-route bridging between 4/16 megabit Token Ring cards in the same Cisco router/bridge. By default, the system enables fast switching in the source-route bridging software. The **no** form of the command restores the default: disabled.

[no] source-bridge spanning

Manually changes the forwarding state of spanning explorer packet. The **no** form of the command disables forwarding.

Source-Bridge Access List and Filtering Commands

[no] access-expression {in|out} *expression*

Interface subcommand that defines an access expression for a given interface—for Token Ring only.

in or **out**—Use **in** to specify that the expression applies to packets entering the interface. Use **out** to specify that the expression applies to packets leaving the interface.

The argument *expression* is a Boolean access list expression, containing one or more of the following:

lsap(nnn)—LSAP access list *nnn* to be evaluated for this frame.

type(nnn)—SNAP type access list to be evaluated for this frame.

smac(nnn)—Access list to match the source MAC address of the frame.

dmac(nnn)—Access list to match the destination MAC address of the frame.

netbios-host(name)—NetBIOS-host access list to be applied on NetBIOS frames traversing the interface.

netbios-bytes(name)—NetBIOS-bytes access list to be applied on NetBIOS frames traversing the interface.

Example:

```
access-expression in lsap(201 | (lsap(202) & dmac(701))
```

[no] access-list *list* { **permit** | **deny** } *type-code wild-mask*

Global configuration command that configures the access list mechanism for filtering frames by protocol type.

list—Access list number.

permit —Specify permit to permit the frame; specify deny to deny the frame.

type-code—LSAP type code for 802-encapsulated packets, or SNAP type code for SNAP-encapsulated packets. 16-bit hexadecimal number.

wild-mask—16-bit hexadecimal number whose ones bits correspond to the bits in the type-code mask that the router should ignore when making a comparison.

Example:

```
access-list 201 permit 0xE0E0 0x0101
```

[no] netbios access-list bytes *name* {**permit** | **deny**} *offset pattern*

Global configuration command that defines the offset and patterns with which to match byte offsets in NetBIOS packets.

name—Name of the access list being defined.

offset—Decimal number indicating the number of bytes into the packet where the byte comparison should begin. An offset of zero points to the beginning of the NetBIOS delimiter string (0xffef) at the start of each NetBIOS packet.

pattern—Hexadecimal string of digits representing a byte pattern.

Example:

```
netbios access-list bytes marketing permit 3 0xabcd
```

[no] netbios access-list host *name* {**permit** | **deny**} *pattern*

Global configuration command that assigns the name of the access list to a station or set of stations on the network. The **no** form of the command removes an entire list, or the entry specified with the *pattern* argument.

name—Name of the access list being defined.

permit or **deny**—Specifies the permit or deny condition.

pattern—Set of characters representing the name of the station, or a combination of characters and pattern matching symbols that establish a pattern for a set of NetBIOS station names.

Example:

```
netbios access-list host marketing permit ABCD
```

[no] netbios input-access-filter bytes *name*

Interface subcommand that defines or removes an access list filter on incoming messages.

name—Name of a NetBIOS access filter.

Example:

```
netbios input-access-filter bytes marketing
```

[no] netbios input-access-filter host *name*

Interface subcommand that defines or removes an access list filter on incoming messages.

name—Name of a NetBIOS access filter.

Example:

```
netbios input-access-filter host marketing
```

[no] netbios output-access-filter bytes *name*

Interface subcommand that defines or removes an access list filter on outgoing messages.

name—Name of a NetBIOS access filter.

Example:

```
netbios output-access-filter bytes marketing
```

[no] netbios output-access-filter host *name*

Interface subcommand that defines an access list filter on outgoing messages.

name—Name of a NetBIOS access filter.

Example:

```
netbios output-access-filter host marketing
```

[no] source-bridge input-address-list *list*

Interface subcommand that assigns an access list to a particular interface for filtering the Token Ring or IEEE 802 source addresses. The **no** version of this command removes the application of the access list.

list—Access list number.

Example:

```
source-bridge input-address-list 201
```

source-bridge input-lsap-list *list*

Interface subcommand that alters IEEE 802-encapsulated packets on input. This access list is applied to all IEEE 802 frames received on that interface prior to the source-routing process. Specify the value zero to disable the filter.

list—Access list number.

Example:

```
source-bridge input-lsap-list 201
```

source-bridge input-type-list *list*

Interface subcommand that filters SNAP-encapsulated packets on input. This access list is then applied to all SNAP frames received on that interface prior to the source-routing process. Specify the value zero to disable the filter.

list—Access list number.

Example:

```
source-bridge input-type-list 201
```

[no] source-bridge output-address-list *list*

Interface subcommand that assigns or removes an access list to or from a particular interface for filtering the Token Ring or IEEE 802 destination addresses.

list—Access list number.

Example:

```
source-bridge output-address-list 201
```

source-bridge output-lsap-list *list*

Interface subcommand that filters IEEE 802-encapsulated packets on output. This access list is then applied just before sending out a frame to an interface. Specify the value zero to disable the filter.

list—Access list number.

Example:

```
source-bridge output-lsap-list 201
```

source-bridge output-type-list *list*

Interface subcommand that filters SNAP-encapsulated on output. This access list is then applied just before sending out a frame to an interface. Specify the value zero to disable the filter.

list—Access list number.

Example:

```
source-bridge output-type-list 201
```

STUN Global Configuration Commands

[no] stun peer-name *ip-address*

Enables or disables STUN.

ip-address—IP address by which this STUN peer is known to other STUN peers that are using the TCP transport.

Example:

```
stun peer-name 131.108.254.6
```

[no] stun poll-interval *milliseconds*

Changes the interval between each sequence of proxy polls generated on the secondary side of the connection. The **no** form of the command restores the default: 20 milliseconds, or 1/50th of a second.

milliseconds—Number of milliseconds in the interval. The minimum value is 20.

Example:

```
stun poll-interval 100
```

[no] stun primary-pass-through *seconds*

Changes the interval between each pass through of polls exchanged by the primary and secondary SDLC devices. The **no** form of the command restores the default: 20 seconds.

seconds—Number of seconds in the interval.

Example:

```
stun primary-pass-through 20
```

[no] stun protocol-group *group-number protocol*

Associates or removes group numbers with protocol names.

group-number—Number from 1 to 255.

protocol—A predefined protocol (**basic** or **sdlc**), or a protocol that you define.

[no] stun schema *name offset constant-offset length address-length*
format *format-keyword*

Specifies a format, or schema, for a user-defined protocol. The **no** form of the command removes the schema.

name—Name of the protocol.

constant-offset—Constant offset, in bytes, for the address to be found in the frame.

address-length—Length of that offset.

format-keyword—Format to be used to specify and display addresses for routes on interfaces that use this STUN protocol. The allowable formats and their maximum lengths are listed in the following table.

Formats	Base	Length
Decimal	Base 10 addresses (0–9)	4
Hexadecimal	Base 16 addresses (0–F)	8
Octal	Base 8 addresses (0–7)	4

Example:

```
stun schema new-sdlc offset 0 length 1 format octal
```

STUN Interface Subcommands

encapsulation stun

Enables the STUN function. You must issue this command to use STUN.

[no] stun group *group-number*

Places a STUN-enabled interface in a defined group.

group-number—Any number from 1 to 255.

Example:

```
stun group 20
```

[no] stun proxy-poll address *address modulus modulus*
{**primary** | **secondary**}

[no] stun proxy-poll address *address discovery*

Enable or disable proxy polling. Default: disabled.

address—Address of the device on which to enable proxy polling.

modulus—Modulus of the link as defined by the MODULUS parameter specified in the line descriptions on the SDLC host.

primary or **secondary**—Indicates whether the SDLC device is primary or secondary.

discovery—Use this keyword if you do not specify primary and secondary ends and connections are negotiated.

Examples:

```
stun proxy-poll address C1 modulus 8 secondary
stun proxy-poll address C1 discovery
```

[no] stun route all tcp *ip-address*
[no] stun route all interface serial *interface-number*
[no] stun route all interface serial *interface-number* **direct**
[no] stun route address *address-number* **tcp** *ip-address*
[no] stun route address *address-number* **interface serial** *interface-number*
[no] stun route address *address-number* **interface serial** *interface-number*
direct

Enable or disable forwarding of frames on the interface.

all—Specifies that all STUN traffic received on the input interface will be propagated, regardless of the address contained in the SDLC frame.

tcp—Causes the TCP transport mechanism to be used to propagate frames that match the entry.

ip-address—Address that identifies the remote STUN peer that is connected to the far SDLC link.

interface serial—Indicates that the Serial Transport method of the STUN function will be used to propagate the SDLC frame.

interface-number—Serial line number connected to the Cisco router.

address—Specifies how an SDLC frame that contains a particular address is to be propagated.

address-number—Address in the SDLC frame, expressed as an octal, decimal, or hexadecimal address in the range allowed by the protocol.

direct—Indicates that the specified interface is a direct STUN link, not a serial connection to another peer.

Examples:

```
stun route address 7 interface serial 1
stun route address 10 tcp 131.108.8.1
stun route all tcp 131.108.10.1
stun route address C1 tcp 131.108.1.1
```

